VAXcluster Console System

digital

Software Manual

Order Number: AA–GV45D–TE

VAXcluster Console System

Software Manual

AA–GV45D–TE

April 1990

This document describes how to use the VAXcluster Console System (VCS). VCS provides a central location from which system managers and operators can perform console functions on multiple systems.

Revision/Update Information:	This manual is a revision replacing the manual of the same name, order number AA–GV45C–TE.
Operating System and Version:	VMS Version 5.3
Software Version:	VCS Version 1.3

digital equipment corporation
maynard, massachusetts

First Printing, May 1986
Revised, May 1987
Revised, November 1988
Revised, April 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Any software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. No responsibility is assumed for the use or reliability of software or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

© Digital Equipment Corporation 1986, 1987, 1988, 1990.

All rights reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation. The following are trademarks of Digital Equipment Corporation: CI, DECnet, DECserver, DECtalk, DECwindows, HSC, MicroVAX, PDP, PDS, RA, SA, TA, TPU, VAX, VAXcluster, VAXstation, VMS, VT, and the DIGITAL logo.

This document was prepared with VAX DOCUMENT, Version 1.2.

Contents

About This Guide xiii

1 Introduction to VAXcluster Console System

1.1 Product Overview 1–1
1.2 Software Components 1–3
1.2.1 I/O Data Logger 1–5
1.2.2 Data Scanner 1–5
1.2.3 Event Logger 1–5
1.2.4 Central Control Coordinator Interface 1–5
1.2.5 Monitor Interface 1–5
1.2.6 Connect Interface 1–5
1.2.7 Record Interface 1–6
1.2.8 Review Interface 1–6
1.2.9 Access Interface 1–6
1.2.10 Configuration Editor 1–6
1.2.11 Configuration File 1–6
1.2.12 Log Files 1–6

2 Central Control Coordinator

2.1 Starting the C^3 2–1
2.2 C^3 Window 2–2
2.2.1 Selecting an Icon 2–4
2.2.2 Choosing a Command 2–4
2.3 Viewing Events 2–5
2.3.1 Scan Events 2–5
2.3.2 Event Information 2–6
2.3.3 VCS Events 2–7

2.4 Connecting to a Node 2–8
2.5 Starting the Monitor Interface 2–9
2.6 Getting Information About the Configuration 2–10
2.7 Reviewing Log Data 2–10
2.8 Setting Host to a Node 2–12
2.9 Customizing C³ General Features 2–13
2.10 Customizing the Connect Interface 2–15
2.11 Customizing the Monitor Interface 2–16
2.12 Customizing the Review Interface 2–17
2.13 Customizing the Set Host Command 2–19
2.14 Saving and Restoring Customization Settings 2–20
2.14.1 Saving Customization Settings 2–20
2.14.2 Restoring Customization Settings 2–21
2.14.3 Customizing C³ DECterm Windows 2–22
2.15 Exiting from the C³ 2–22

3 Creating the VCS Configuration with the Configuration Editor DECwindows Interface

3.1 Configuration File 3–1
3.2 Starting the Configuration Editor 3–3
3.3 Adding a Node 3–4
3.4 Adding a Peripheral 3–9
3.5 Adding a Cluster 3–11
3.6 Adding an Event 3–13
3.7 Adding a Scan Profile 3–16
3.8 Adding a VCS User 3–18
3.9 Setting Up the VCS Environment 3–20
3.10 Labels 3–23
3.11 Modifying the Configuration 3–24
3.11.1 Adding Duplicate Nodes 3–25
3.11.2 Deleting from the Configuration 3–25
3.12 Customizing the Central Control Coordinator 3–25
3.13 Verifying the Configuration 3–26
3.14 Exiting from the Configuration Editor 3–26

4 Configuration Editor Commands

@ (EXECUTE COMMAND FILE) 4–2
ADD 4–3
ADD CLUSTER 4–4
ADD EVENT 4–6
ADD NODE 4–12
ADD PERIPHERAL 4–17
ADD SCAN 4–19
ADD USER 4–22
DELETE 4–25
EXIT 4–27
HELP 4–28
MODIFY 4–30
MODIFY CLUSTER 4–31
MODIFY EVENT 4–33
MODIFY NODE 4–36
MODIFY PERIPHERAL 4–39
MODIFY SCAN 4–41
MODIFY USER 4–43
QUIT 4–45
SET 4–46
SET DISPLAY 4–47
SET LOGGING 4–49
SET SECURITY 4–52
SET VERIFY 4–53
SHOW 4–54
SPAWN 4–57
VERIFY 4–59

5 DCL Commands

5.1 Node Name Conventions 5–1
5.2 DCL Command Reference 5–2
VCS 5–3
VCS ACCESS 5–4
VCS ACCESS /DEFAULT 5–7
VCS CONNECT 5–10
VCS CONVERT 5–15

VCS COORDINATE 5–19
VCS EDIT CONFIGURATION 5–22
VCS MONITOR 5–24
VCS RECONFIGURE 5–27
VCS RECORD 5–29
VCS REVIEW 5–32
VCS SHUTDOWN 5–38
VCS START LOGGER 5–39
VCS START SCANNER 5–40

6 Monitor Interface

6.1 Introduction 6–1
6.1.1 Connect Mode 6–2
6.1.2 Monitor Mode 6–2
6.1.2.1 Log Window 6–4
6.1.2.1.1 Event Log Window 6–6
6.1.2.1.2 Event Context Window 6–7
6.1.2.2 Event Summary Window 6–7
6.1.2.3 Command Line 6–9
6.1.2.4 Message Line 6–9
6.2 Command Processing in the Monitor Mode 6–9
6.2.1 Line Editing Functions 6–9
6.2.2 Key Definitions 6–10
6.2.3 VAXstation Mouse 6–12
6.2.4 Node Names 6–12
6.2.5 Text Strings Contained in Command Lines 6–12
6.2.6 Specifying Control Characters 6–13
6.2.7 Date and Time Arguments 6–13
6.3 Monitor Interface Commands 6–14
@ (EXECUTE COMMAND FILE) 6–15
CONNECT 6–17
CYCLE 6–22
DEFINE/KEY 6–23
DISABLE 6–28
ENABLE 6–30
EXIT 6–32
FIND 6–33
GOTO 6–36
HELP 6–38
NEXT EVENT 6–41

NEXT SCREEN 6–43
ONE_WINDOW 6–44
OTHER_WINDOW 6–45
OUTPUT 6–46
PREVIOUS EVENT 6–49
PREVIOUS SCREEN 6–51
RECONFIGURE 6–52
REFRESH 6–54
REVIEW 6–55
SAVE 6–61
SCROLL 6–63
SELECT 6–65
SET BELL 6–68
SET BREAK 6–69
SET ESCAPE 6–71
SET LOCK 6–73
SET MOUSE 6–75
SET PRIVILEGES 6–76
SET SHIFT_KEY 6–77
SET TIME_STAMP 6–79
SET WIDTH 6–80
SHIFT 6–81
SHOW DIRECTORY 6–83
SHOW KEYS 6–85
SHOW MONITOR 6–87
SHOW NODES 6–89
SHOW TIME 6–92
SHUTDOWN 6–93
TWO_WINDOWS 6–95
UNDEFINE/KEY 6–96
VIEW 6–98
WAIT 6–101
WRITE 6–103

7 Summary of Monitor Interface Tasks

7.1 Entering and Exiting the VCS Monitor Interface 7–3
7.2 Getting Help 7–4
7.3 Communicating with Nodes 7–5
7.4 Viewing Current Console Logs 7–6
7.5 Viewing Events 7–6
7.6 Viewing Event Context 7–7
7.7 Controlling the Window Display 7–7
7.8 Searching for Information 7–8
7.9 Showing Information About the Current Monitor Interface Session 7–9
7.10 Reviewing Log Data 7–9
7.11 Copying a Buffer into a File 7–10
7.12 Viewing the Messages Received in the Current Monitor Interface Session 7–10
7.13 Modifying Key Definitions 7–10
7.14 Tailoring Your Monitor Interface Environment 7–11

8 Creating Your Own Icons

8.1 Creating an Icon Picture 8–3
8.1.1 Drawing an Icon Picture 8–3
8.1.2 Capturing an Icon Picture 8–3
8.2 Converting the Image 8–4
8.3 Reviewing the Icon 8–5
8.3.1 Getting Information About an Icon 8–6
8.3.2 Icon Names 8–7
8.3.3 Icons Supplied by VCS 8–7
8.3.4 Exiting from the Icon Review 8–11
8.4 Using the Icon 8–12

9 Writing Access Interface Applications

9.1 Overview of the Callback Routines 9–2
9.2 Creating User Code 9–3
9.3 Application Design Considerations 9–11
9.3.1 Event Management 9–12
9.3.2 Testing Environment 9–12
9.3.3 Logic Verification 9–13
9.4 Linking and Executing User Code 9–14
9.5 Debugging User Code 9–15
9.6 Customizing the DCL Command Syntax 9–17

9.7 Callback Routine Reference 9–17
VCS$USER_INITIALIZE 9–19
VCS$USER_MAINLOOP 9–22
VCS$USER_RECONFIGURE 9–24
VCS$USER_SCAN_EVENT 9–26
VCS$USER_SHUTDOWN 9–29
VCS$USER_VCS_EVENT 9–30

10 Event Notification System

10.1 Getting Started 10–2
10.1.1 Starting ENS 10–2
10.1.2 Using the ENS Editor 10–2
10.1.3 Testing ENS 10–4
10.1.4 Log Files 10–4
10.1.5 Logical Names 10–4
10.2 ENS Action Routines 10–4
10.2.1 Brdcst 10–5
10.2.2 Mail 10–5
10.2.3 DECtalk 10–6
10.2.4 Window 10–7
10.2.4.1 Window Customization 10–8
10.2.5 Window1 10–10
10.2.5.1 Window1 Customization 10–11
10.3 Adding Your Own Action Routines 10–12
10.3.1 Command Procedure Action Routines 10–13
10.3.2 Image Action Routines 10–13
10.3.2.1 Using Mailboxes 10–14
10.3.2.2 Passing a Parameter String to an Image 10–15
10.3.2.3 Passing a Parameter Structure to an Image 10–15
10.3.2.4 Testing and Debugging an Action Routine 10–17
10.4 ENS Editor Commands 10–18
ADD 10–19
ADD ACTION 10–20
ADD EVENT 10–24
COPY 10–26
COPY ACTION 10–27
COPY EVENT 10–30
DELETE 10–32
EXIT 10–33
HELP 10–34
MODIFY 10–36

MODIFY ACTION 10–37
MODIFY EVENT 10–39
SHOW 10–41

A Log Files

A.1 Introduction A–1
A.2 Log File Analysis A–3
A.3 Console and Event Log File Record Formats A–3
A.3.1 Version Record A–6
A.3.2 VCS Error Record A–8
A.3.3 Service Initiation/Termination Record A–10
A.3.4 User Status Change Record A–11
A.3.5 User to Node Output Record A–13
A.3.6 Log Data Record — (Partial, Complete, or Full) A–14
A.3.7 VCS Event Record A–16
A.3.8 Scan Event Record A–18

B System Messages and Error Recovery Procedures

B.1 General Instructions B–1
B.2 System Messages and Error Recovery Procedures B–2

Index

Examples

9–1 Sample Application Code 9–3

Figures

1–1 VCS Console Data Flow 1–4
4–1 Event Anatomy 4–9
5–1 Recorded Text Anatomy 5–30
5–2 Reviewed Text Anatomy 5–35
5–3 Reviewed Event Anatomy 5–36
6–1 Monitor Interface Display with One Log Window 6–3
6–2 Monitor Interface Display with Two Log Windows 6–5
6–3 Event Summary Window 6–8

6–4 Window Control Keys 6–11
6–5 Keypad Layout 6–25
6–6 Reviewed Text Anatomy 6–58
6–7 Reviewed Event Anatomy 6–59
8–1 Creating Your Own Icons 8–2
9–1 VCS Access Interface Argument Block 9–18
10–1 ENS Window 10–7
10–2 ENS Window Action Routine Icon 10–8
10–3 ENS Window1 10–10
10–4 ENS Mailbox Structure 10–16
A–1 Record Header A–3
A–2 Version Record A–6
A–3 VCS Error Record A–8
A–4 Service Initiation/Termination Record A–10
A–5 User Status Change Record A–11
A–6 User to Node Output Record A–13
A–7 Log Data Record A–14
A–8 VCS Event Record A–16
A–9 Scan Event Record A–18

Tables

2–1 Icon Status 2–2
4–1 Configuration Editor Commands 4–1
4–2 Node Access Commands 4–23
4–3 VCS Management Commands 4–24
5–1 VCS DCL Commands 5–2
6–1 Command Line Editing Keys 6–10
6–2 Monitor Interface Commands 6–14
6–3 Control Key Names 6–23
6–4 Reserved Control Key Names 6–24
6–5 Function and Keypad Key Names 6–24
7–1 Monitor Interface Tasks and Commands 7–1
9–1 Access Interface Callback Routines 9–2
9–2 VCS Common Source Files 9–3
9–3 VCS$USER_INITIALIZE Argument List 9–20
9–4 VCS$USER_MAINLOOP Argument List 9–22
9–5 VCS$USER_RECONFIGURE Argument List 9–25
9–6 VCS$USER_SCAN_EVENT Argument List 9–26

9–7 VCS$USER_VCS_EVENT Argument List 9–30
9–8 VCS Event Codes 9–31
10–1 ENS Window Routine Customization Parameters 10–8
10–2 ENS Window1 Routine Customization Parameters 10–11
10–3 ENS Parameter Names and Definitions 10–12
10–4 ENS Mailbox Message Types 10–14
A–1 Record Header Fields A–4
A–2 Console and Event Log Record Types A–4
A–3 Version Record Information A–7
A–4 VCS Error Record Information A–9
A–5 Service Initiation/Termination Record Information A–10
A–6 User Status Change Record Information A–12
A–7 User Change Codes A–12
A–8 User to Node Output Record Information A–13
A–9 Log Data Record Information A–15
A–10 Partial Lines A–15
A–11 VCS Event Record Information A–17
A–12 VCS Event Codes A–17
A–13 Scan Event Record Information A–19
B–1 Severity Levels B–1

About This Guide

Document Structure

Chapter 1, Introduction to VAXcluster Console System, provides an overview and the basic concepts of the VAXcluster Console System (VCS).

Chapter 2, Central Control Coordinator, explains how to use the Central Control Coordinator (C^3) operator interface.

Chapter 3, Creating the VCS Configuration with the Configuration Editor DECwindows Interface, describes how to set up the VCS environment with the VCS Configuration Editor.

Chapter 4, Configuration Editor Commands, is a reference to the Configuration Editor line-mode commands.

Chapter 5, DCL Commands, lists the DCL commands, qualifiers, and keywords. Commands are listed in alphabetical order.

Chapter 6, Monitor Interface, explains how to use the Monitor Interface. The second part of the chapter is a reference for the Monitor Interface commands.

Chapter 7, Summary of Monitor Interface Tasks, presents a task-oriented summary of Monitor Interface and DCL commands.

Chapter 8, Creating Your Own Icons, describes how to create your own icons for the Configuration Editor and Central Control Coordinator.

Chapter 9, Writing Access Interface Applications, describes how to create your own application for use with the VCS Access Interface.

Chapter 10, Event Notification System, describes the Event Notification System supplied with VCS.

Appendix A, Log Files, describes the record format of the log files.

Appendix B, System Messages and Error Recovery Procedures, lists VCS system messages and error recovery procedures.

Intended Audience

This manual addresses the needs of four groups of VCS users:

- Operators who use VCS to monitor, initiate, and respond to events occurring on a serviced node.
- System managers who use VCS to manage multiple systems from a central location.
- Digital Customer Services and software support personnel who use VCS to diagnose problems on a serviced node.
- System programmers who write application software for VCS.

Associated Documents

The following documents are either referenced in this manual or contain additional information pertaining to VCS:

DECwindows User Guide
VAXcluster Console System Installation Guide
Digital VAX-11 Software Engineering Manual
VAX/VMS System Messages and Recovery Procedures
VAX/VMS Linker Reference Manual
VAX/VMS Command Definition Utility Reference Manual
VMS DECwindows Guide to Application Programming
Guide to VMS System Security
Introduction to VMS System Management
Introduction to VMS System Routines
VMS DCL Concepts Manual
VMS System Manager's Manual

Revision History

The following list describes the major documentation differences between VCS Version 1.2 and Version 1.3.

- Chapter 1, Introduction to VAXcluster Console System, is similar to the Version 1.2 manual.
- Chapter 2, Central Control Coordinator, is a new chapter. The C^3 replaces the Version 1.2 Layout Application. Chapter 2 replaces Version 1.2 Section 8.2, Layout Application.
- Chapter 3, Creating the VCS Configuration with the Configuration Editor DECwindows Interface, is a new chapter. The Configuration Editor DECwindows interface replaces the Version 1.2 Layout Editor. Chapter 3 replaces the following Version 1.2 information:
  - Chapter 7, Using the Layout Editor
  - Appendix B, Using the Scan File, Configuration File, and Layout File
- Chapter 4, Configuration Editor Commands, is a new chapter. Chapters 3 and 4 replace parts of Version 1.2 Appendix B, Using the Scan File, Configuration File, and Layout File.
- Chapter 5, DCL Commands, has been updated to reflect the Version 1.3 DCL commands.

  The changed commands are:

  $ VCS ACCESS

  $ VCS EDIT CONFIGURATION

  $ VCS MONITOR

  The new commands are:

  $ VCS CONVERT

  $ VCS COORDINATE

  The obsolete commands are:

  $ VCS ACCESS /LAYOUT (replaced by VCS COORDINATE)

  $ VCS EDIT LAYOUT (replaced by VCS EDIT CONFIGURATION

  $ VCS EDIT SECURITY (replaced by VCS EDIT CONFIGURATION)
- Chapter 6, Monitor Interface, is similar to the Version 1.2 manual.

- Chapter 7, Summary of Monitor Interface Tasks, has not been changed.
- Chapter 8, Creating Your Own Icons, is a new chapter. The ability to create custom icons is a new function for Version 1.3.
- Chapter 9, Writing Access Interface Applications, has been expanded with more guidelines for programming VCS applications.
- Chapter 10, Event Notification System, is a new chapter. The Event Notification System is a new set of applications for Version 1.3.
- Appendix A, Log Files, is similar to the Version 1.2 manual.
- Appendix B, System Messages and Error Recovery Procedures, has been updated to reflect the Version 1.3 error messages.

Conventions

Italic	Words or phrases appearing in italics indicate one of the following: ■ Referencing of an associated document ■ Emphasis on an important word or concept ■ A variable
mouse	Refers to any pointing device, such as a mouse, a puck, or a stylus.
MB1 and MB2	MB1 indicates the left mouse button. MB2 indicates the middle mouse button. (The buttons can be redefined by the user.)
Blue-green ink	Blue-green ink indicates information that you must enter from the keyboard or a screen object that you must choose or click on.
UPPERCASE TEXT	Text in uppercase should be typed exactly as shown.
lowercase text	Text in lowercase is an item supplied by you, such as a file name or a node name.
Return	In procedures, a key name is shown enclosed to indicate that you press a key on the keyboard.

Ctrl/x	In procedures, a sequence such as Ctrl/x indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
[ ]	In format descriptions, brackets ([ ]) enclose an optional item.
{ }	In format descriptions, braces ({ }) indicate that you must choose one item from the enclosed list.
, . . .	A comma followed by a horizontal ellipsis indicates that you can supply more than one item separated by commas.
. . .	Vertical ellipses indicate that irrelevant portions of a programming example are excluded for brevity.
Boldface text	Boldface is used to indicate the following: ■ User input in online versions of this book ■ System messages or text strings discussed within other text

1

Introduction to VAXcluster Console System

This chapter gives you an overview of the VAXcluster Console System (VCS) product and its software components.

1.1 Product Overview

VCS is a VMS Version 5.3 DECwindows layered software product. Coupled with a VAX, MicroVAX, or VAXstation host system and other appropriate hardware, VCS provides a central point for the coordination of system console services and for the logging of all console data received from the serviced nodes. Although VCS may function on any VAX hardware configuration that supports VMS Version 5.3, only those configurations that are listed in the Software Product Description have been tested and are supported.

Previously, a system console terminal was the only location from which you could perform console functions for that system. Now, when you connect a system console to VCS, you can perform console functions for that system from a terminal or VAXstation display connected to the VCS host system. If you connect a number of system consoles to VCS, you can still perform console functions for all the serviced systems from that single location. VCS eliminates the need for an operator to monitor physically separate system consoles that are often located at different places at a single site.

VCS allows you to connect to and service the console devices of as many as 32 serviced nodes. These serviced nodes can be:

- VAXcluster systems
- Standalone VAX systems

- Any other device that meets the following requirements:
  - Sends ASCII data over an RS-232C line
  - Has an EIA console port
  - Supports XON/XOFF and I/O buffering

Hence, PDP-11s, line printer servers or controllers, LAN bridges, and other third-party devices that meet the above requirements can be serviced by VCS.

The VCS host system connects to the console port of each serviced node with an asynchronous link. Depending on the need for signal protection, the asynchronous link can be composed of DECconnect or fiber optic components. One end of the link is terminated at the console port of the serviced node. The other end is terminated at the VCS host system by one of the following methods:

- Direct connection, such as connecting to a port of a serial line interface device (such as DHV11, DHQ11, or DZQ11) within the VCS host system cabinet.
- LAT host initiated connection, such as connecting to a port of a DECserver 200 or 300 terminal server which is accessible by the VCS host system over the Ethernet. The terminal server port must be mapped to a host initiated connection LTA terminal device defined in the VCS host system.

From a single terminal or VAXstation display, connected to the VCS host system, you can perform all console functions for all serviced nodes. These functions include but are not necessarily limited to:

- Shutting down or rebooting the serviced nodes
- Running standalone diagnostics
- Performing standalone backup operations
- Installing layered products
- Viewing console output
- Reviewing historical console data
- Retrieving historical console data for printing or analysis
- Searching for console data

In addition to being a central point for the coordination of system console services, VCS performs the following functions:

- Logs all data and activities between VCS and the serviced nodes.
- Scans incoming messages from the serviced nodes for specific text strings that may contain status or error information.
- Enables you to assemble icons (not drawn to scale) into graphics displays on a VAXstation screen representing the floor plan of your data center and the logical grouping of your VAXcluster configurations.
- Provides an optional security facility to control access to the serviced nodes.

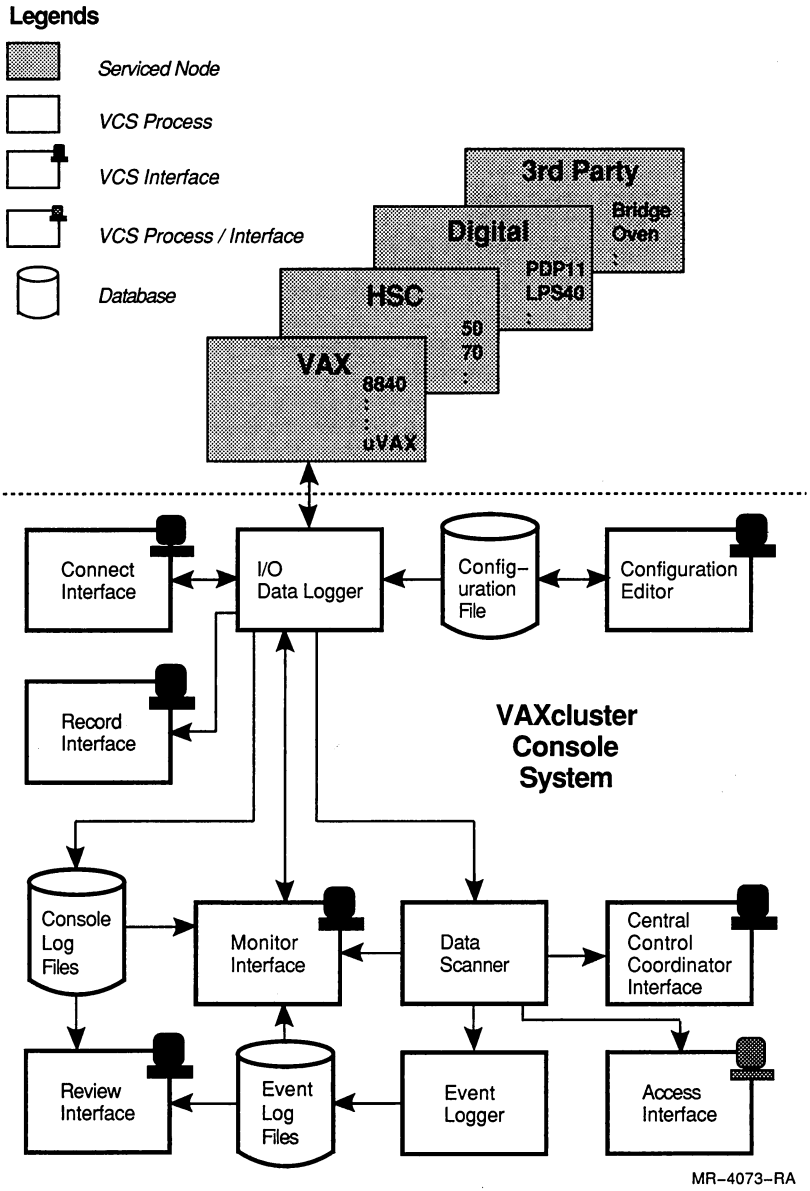
1.2 Software Components

VCS requires the following software:

- VMS Version 5.3
- DECwindows

The following sections briefly describe the VCS software components and their functions. Figure 1–1 shows the relationship between the VCS components, and the console data flow among them.

Figure 1–1 VCS Console Data Flow



1.2.1 I/O Data Logger

The I/O Data Logger process manages the console lines and logs all data received on these lines. In addition, the I/O Data Logger makes the received data available to other VCS components and transmits the data on the console lines to nodes designated by those components.

1.2.2 Data Scanner

The Data Scanner process scans the console log data for predefined events. The information about detected events is relayed to other VCS components for logging and additional processing.

1.2.3 Event Logger

The Event Logger process records events detected by the Data Scanner. The Event Logger logs all events from all currently configured systems to a disk log file.

1.2.4 Central Control Coordinator Interface

The Central Control Coordinator Interface is a DECwindows interface that displays the floor plan of your data center while allowing you to monitor all console activities and to respond to events detected on the console connections. Console events are reflected by the node icons in the display and direct the operator to systems needing attention. Other interactive interfaces can also be activated from the Central Control Coordinator Interface.

1.2.5 Monitor Interface

The Monitor Interface enables you to monitor the serviced nodes by displaying multiple windows of information. You can connect the primary window to a single system or display log data in this window while monitoring other nodes.

1.2.6 Connect Interface

The Connect Interface enables you to connect to a serviced node, making your terminal the dedicated console. (More than one operator can connect to the same node.) Once you are connected, all keystrokes (except for defined *Break* and *Escape* characters) are transmitted directly to the console line of the node to which you are connected. While connected to that node, you have the normal console terminal behavior. The only exception to normal behavior is that one control character is reserved to enable you to escape back to the VCS system from which the connection was made.

1.2.7 Record Interface

The Record Interface enables you to capture console line activity on a hardcopy device.

1.2.8 Review Interface

The Review Interface enables you to retrieve console data and event information from the console and event log files. The retrieved information and data can be specified in terms of source node and time interval.

1.2.9 Access Interface

The Access Interface is a programming interface that provides an alternate method for capturing events for additional processing. You can write your own application for use with the Access Interface.

In addition, VCS provides the Default Application and a DECwindows application, each with source code. These applications are ready-made applications you can use with the Access Interface. The Default Application is a complete utility that sends scan and VCS events to a specified output device. The DECwindows application displays a list of serviced nodes and the name of the highest priority event that has occurred on each.

1.2.10 Configuration Editor

The Configuration Editor helps you create and maintain the configuration information that VCS requires.

1.2.11 Configuration File

The configuration file contains information about the nodes being serviced by VCS and the VCS environment itself. The Configuration Editor is provided to aid in the creation and modification of this information.

1.2.12 Log Files

VCS maintains the console and event log files as the permanent historical records of console data from the serviced nodes. VCS creates a new set of console and event log files every 24 hours, beginning at midnight.

When additional space is needed on the disk device, VCS displays a warning message before disk space shortage is critical. Then, as additional space is required, VCS deletes the oldest log files to make room for the new ones. The system manager can back up the log files before VCS deletes them.

Console Log Files

The I/O Data Logger process collects console data from the serviced nodes and makes the data available to other VCS processes. As the strings of ASCII data are received, they are also written into a log file on disk. The Monitor Interface enables you to view some of the most recent log data on your terminal screen.

The console data from all serviced nodes is logged in chronological order. Using the Review Interface, you can examine the data from a single node from the log files as well as examine the interactions between systems during a specified time period.

Event Log Files

The Event Logger process receives events from the Data Scanner and writes them into an event log file. The Central Control Coordinator and Monitor Interfaces enable you to view all of the events pertaining to the set of serviced nodes that are being monitored, starting at the moment that interface is activated.

The events from all serviced nodes are logged in chronological order. Using the Review Interface, you can examine the events from a single node from the event log files as well as examining the events from more than one system during a specified time period.

2

Central Control Coordinator

The Central Control Coordinator (C^3) is the principal VCS user interface. This DECwindows interface provides you with the overall picture of the managed environment, its serviced entities, and their status.

The C^3 uses the computer room layout information created by the Configuration Editor to provide you with a graphic view of your data center and an interface to VCS functions. From the C^3, you can invoke the following VCS functions:

- Viewing scan and VCS events
- Connecting to a system's console
- Monitoring one or more systems with the Monitor Interface
- Getting information about a system or event
- Reviewing a system's console or event data
- Setting host and logging in to any serviced system or the VCS host system

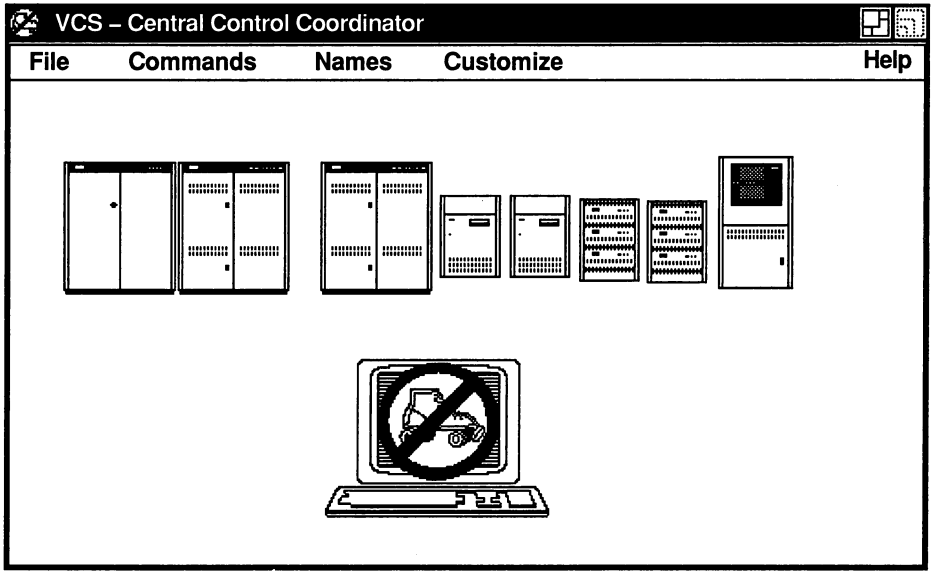
2.1 Starting the C^3

When you start VCS as described in the *VAXcluster Console System Installation Guide*, the C^3 starts automatically if the DECwindows environment is present.

You can also start the C^3 with the VCS COORDINATE command. See Chapter 5 for complete details.

2.2 C³ Window

The C³ displays a window that contains icons for all the nodes and peripherals in the VCS configuration. The icons are located in the same positions in which you placed them with the Configuration Editor. The window opens in the same size and position that you chose with the Configuration Editor.



The C³ alerts you of the most recent scan events by updating the color (or pattern on a monochrome display) of the relevant node icon. The color indicates the priority level of the latest scan event that has occurred on a node. Table 2–1 correlates node status to icon colors.

Table 2–1 Icon Status

Node Status	Icon Color
Event Priority 100	Black
Event Priority 61 — 99	Red
Event Priority 31 — 60	Yellow
Event Priority 1 — 30	Green
No New Events	Wheat

(continued on next page)

Table 2–1 (Cont.) Icon Status

Node Status	Icon Color
Node Disabled to VCS	Gray
No Access to User	Dimmed
Selected for Command	Highlighted

No New Events
A plain icon means that no new events have occurred on the node.

Locked
A lock symbol on the icon means the node's console is locked, that is, unavailable for input from other operators.

Offline Event
A black icon indicates that an operator has issued the DISABLE command to disable VCS services to the node.

High Priority Event
A red (wave patterned on a monochrome display) icon indicates that a high priority event has occurred on the node.

Medium Priority Event
A yellow (diagonal striped on a monochrome display) icon indicates that a medium priority event has occurred on the node.

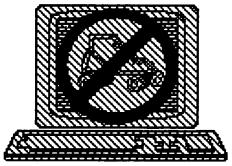
Low Priority Event
A green (dotted on a monochrome display) icon indicates that a low priority event has occurred on the node.

Disabled to VCS
A gray icon means that VCS services to the node are disabled.

Selected
A highlighted icon means that it has been selected for a command.

No Access to User

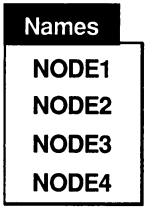
A dimmed icon means that it cannot be selected for a command because its node name was not included on the VCS COORDINATE command line. A peripheral icon is dimmed if all of its parent node icons are dimmed.



New VCS Event

The occurrence of a new VCS event is indicated by diagonal lines across the VCS host icon.

2.2.1 Selecting an Icon



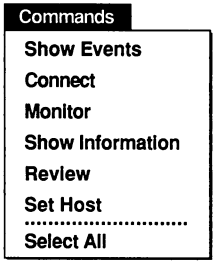
You can select one or more icons on which to perform a command in any of the following ways:

- By pulling down the Names menu and clicking on the name of each icon you want to select.
- By pulling down the Commands menu and choosing Select All.
- By clicking on the icon itself. To select more than one icon, shift click each icon.
- By double-clicking on the node icon itself to view events.

You can select multiple icons for the Show Events, Monitor, and Review commands.

If you change your mind about selecting an icon, click on it again and it will no longer be selected.

2.2.2 Choosing a Command



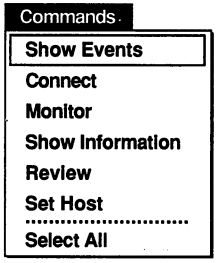
You can choose a command in any of three ways:

- By pulling down the Commands menu.
- By clicking MB2 on any icon to display a pop-up menu.
- By double-clicking MB1 on a node icon itself to view events. Double-clicking on a peripheral icon displays information about the peripheral.

2.3 Viewing Events

A scan event is the receipt of a console message from a serviced node. Using the C^3, you can display the scan events that have occurred on serviced nodes and the VCS events that have occurred on the host system.

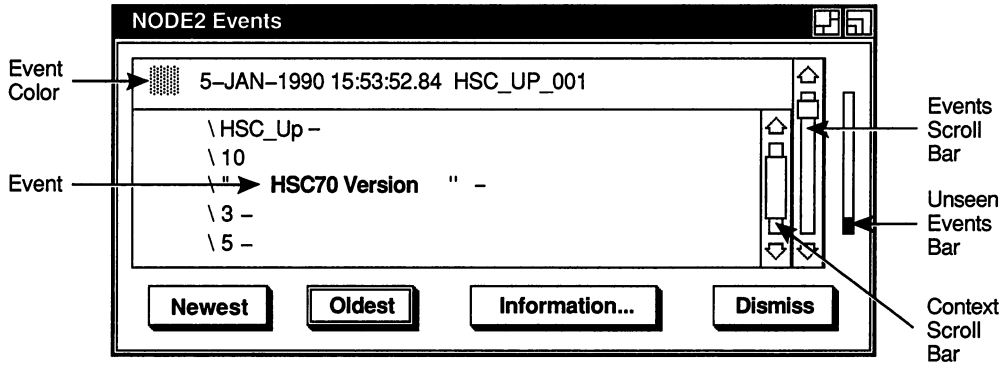
2.3.1 Scan Events



To view scan events, double click on a node icon. You can also select the node icons you want, then choose Show Events from the Commands menu. Or, choose Show Events from the pop-up menu on any node icon.

You can open multiple event windows by shift-clicking on all the node icons you want, then choosing Show Events from the Commands menu.

An event window opens. The window contains the latest event and its log file context. The event is highlighted. If a new event occurs while an event window is open, the display is updated.



The bar on the right side of the event window shows the proportion of unseen events to seen events.

Drag the slider on the events scroll bar to scroll through all events, seen and unseen.

If all the event context does not fit inside the event window, a context scroll bar is displayed. Drag the slider on the context scroll bar to scroll though the context of the event.

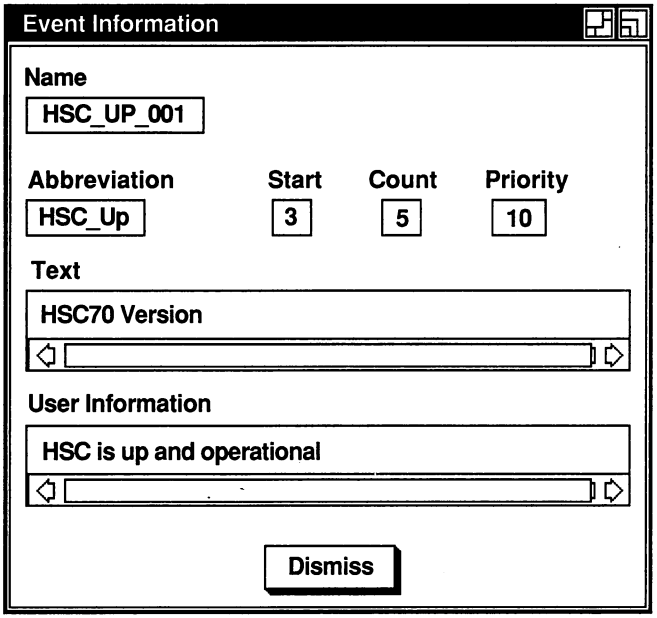
If the log file records containing the scan event context are locked by another user, the event window displays the scan event without context. Only log files from the last two days are kept open. If an event is older than two days, the event window displays only the scan event without context. If a node has logging disabled, no context is displayed.

To display the newest unseen events, click on the Newest button. To display the oldest unseen events, click on the Oldest button. The Newest and Oldest buttons are dimmed if there are no new unseen events.

Click on the Dismiss button to close the event window. The icon will return to wheat color indicating no new events.

2.3.2 Event Information

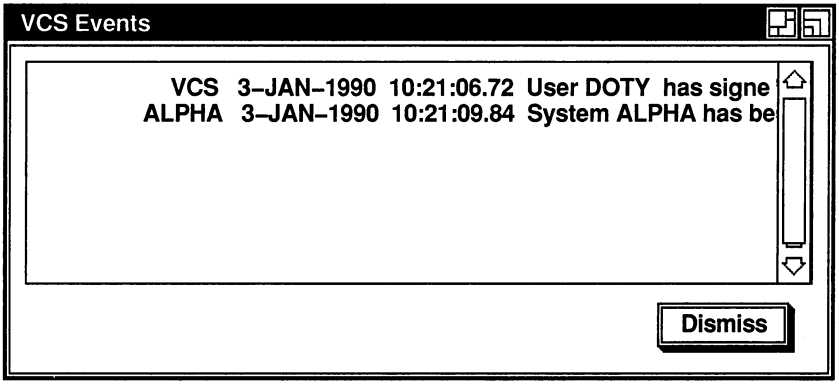
To display information about the event, click on the Information . . . button.



Click on the Dismiss button to close the Event Information window.

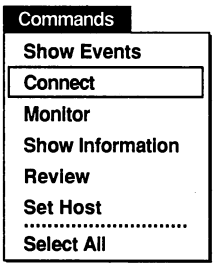
2.3.3 VCS Events

To view VCS events, click on the VCS host icon. The C³ displays the latest VCS events.



Resize the window horizontally, if necessary, to view the full width of the VCS events. Click on the Dismiss button to close the VCS event window.

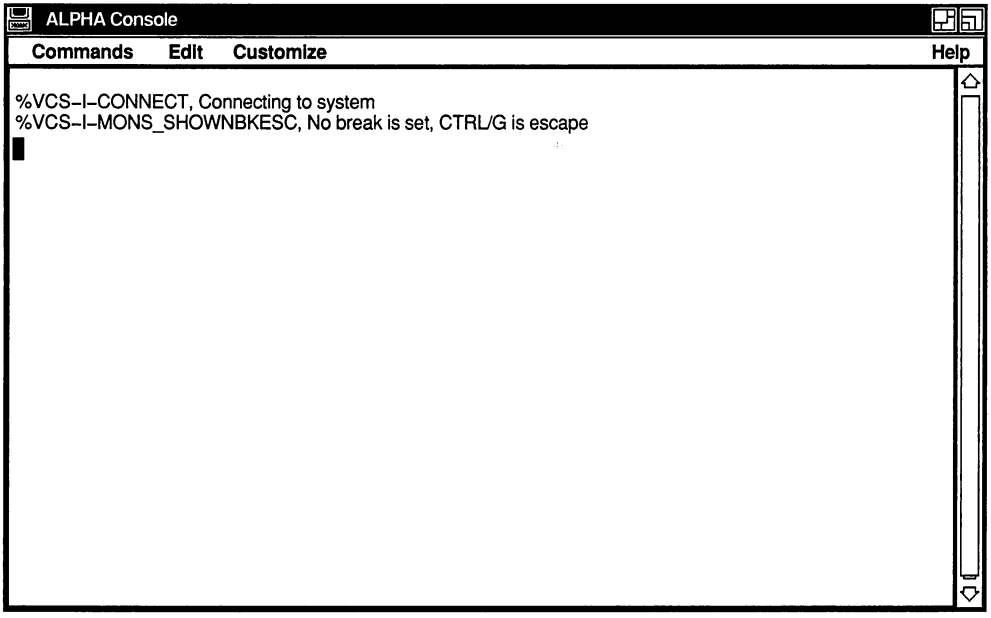
2.4 Connecting to a Node



Connecting to a node allows you to communicate with that node as if you were on a local console terminal. By default, the node is unlocked so that other operators can send data to the same console line to which you are connected.

To connect to a node, select the node icon you want, then choose Connect from the Commands menu. Or, choose Connect from the pop-up menu on any node icon.

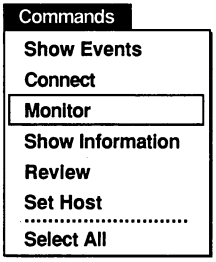
A new terminal window opens. If you have customized the C^3 to lock the console (see Section 2.10), the icon in the C^3 window displays a lock symbol. The window title bar displays ***Node-name*** **Console**. Click on the new window to give it keyboard input focus.



To end the connection, pull down the Commands menu and choose Quit. Or, press Ctrl/G, then confirm by entering Yes.

For more detailed information about connecting to a node, see the Monitor Interface CONNECT command or the VCS CONNECT command.

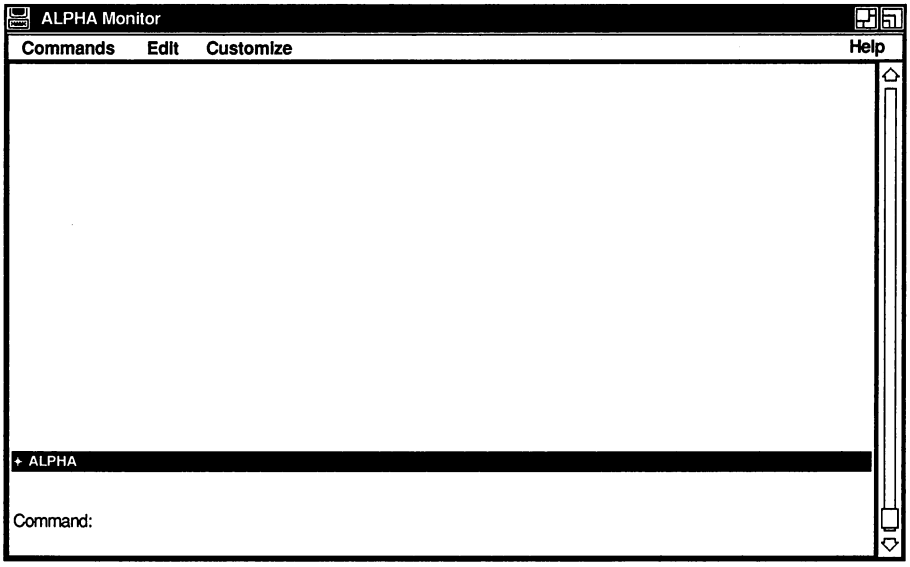
2.5 Starting the Monitor Interface



To start the Monitor Interface, select the node icon you want, then choose Monitor from the Commands menu. Or, choose Monitor from the pop-up menu on any node icon.

You can monitor multiple nodes in one window by shift-clicking on the node icons you want, then choosing Monitor from the Commands menu. Or, if you select the VCS host icon, the Monitor session is for all serviced nodes.

A new terminal window opens. The window title bar displays ***Node-name*[,*Node-name*, . . .] Monitor**. Click on the window to give it keyboard input focus.

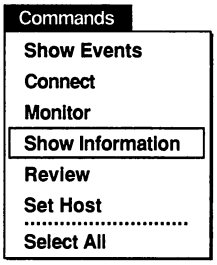


For detailed information about the Monitor Interface, see Chapter 6.

To end the Monitor Interface session, do any of the following:

- Pull down the DECterm Commands menu and choose Quit.
- Enter Exit at the prompt.
- Press F10.

2.6 Getting Information About the Configuration

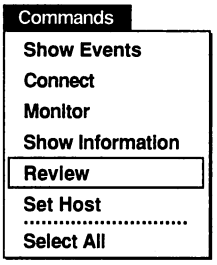


You can display all the information that has been entered into the configuration.

Select the icon you want information about, then choose Show Information from the Commands menu. Or, choose Show Information from a pop-up menu on any icon.

See the Configuration Editor ADD commands in Chapters 3 and 4 for descriptions of the information displayed by the Show Information menu item.

2.7 Reviewing Log Data



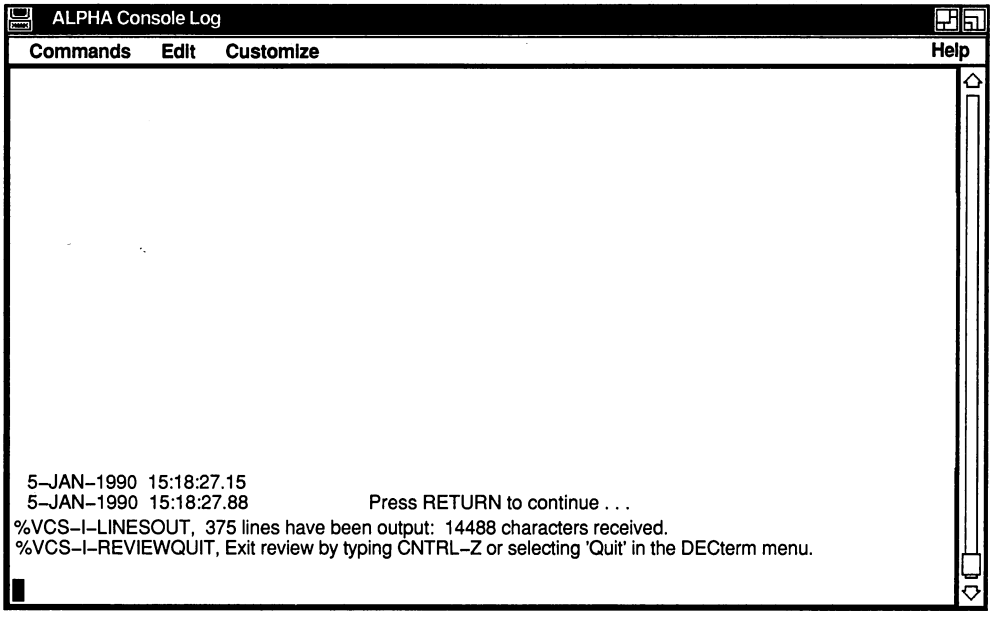
The Review Interface allows you to examine console and event log data. By default, the Review Interface displays console log data. To examine event log data, customize the Review Interface as described in Section 2.12.

To start the Review Interface, select the node icon you want, then choose Review from the Commands menu. Or, choose Review from the pop-up menu on any node icon.

You can review log data for multiple nodes in one window by shift-clicking on the node icons you want, then choosing Review from the Commands menu.

If you select the VCS host icon, the Review Interface displays a review of VCS events.

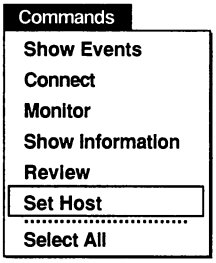
A new terminal window opens. The window title bar displays ***Node-name*[,*Node-name*, . . .] Console Log** (or ***Node-name*[,*Node-name*, . . .] Event Log** if you are examining the event log data). Click on the window to give it keyboard input focus.


ALPHA Console Log
Commands Edit Customize Help
5–JAN–1990 15:18:27.15
5–JAN–1990 15:18:27.88 Press RETURN to continue . . .
%VCS–I–LINESOUT, 375 lines have been output: 14488 characters received.
%VCS–I–REVIEWQUIT, Exit review by typing CNTRL–Z or selecting 'Quit' in the DECterm menu.

The Review Interface displays a new page of log data each time you press Return. To quit reviewing and close the window:

- Press Ctrl/Z
- Choose Quit from the DECterm Commands menu

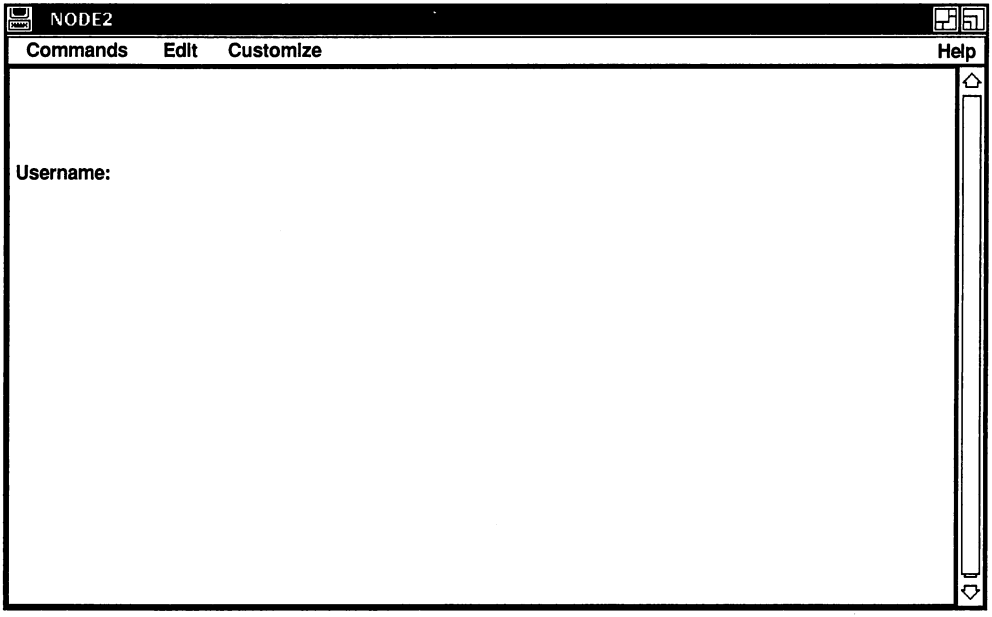
2.8 Setting Host to a Node



To set host to a node, select the node icon you want, then choose Set Host from the Commands menu. Or, choose Set Host from the pop-up menu on any node icon.

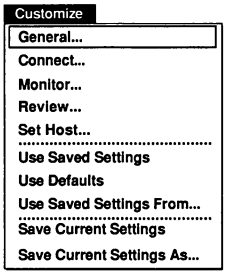
You can set host to the VCS host system as well as to the serviced nodes.

A new terminal window opens. The window title bar displays ***Node-name***. Click on the window to give it keyboard input focus.

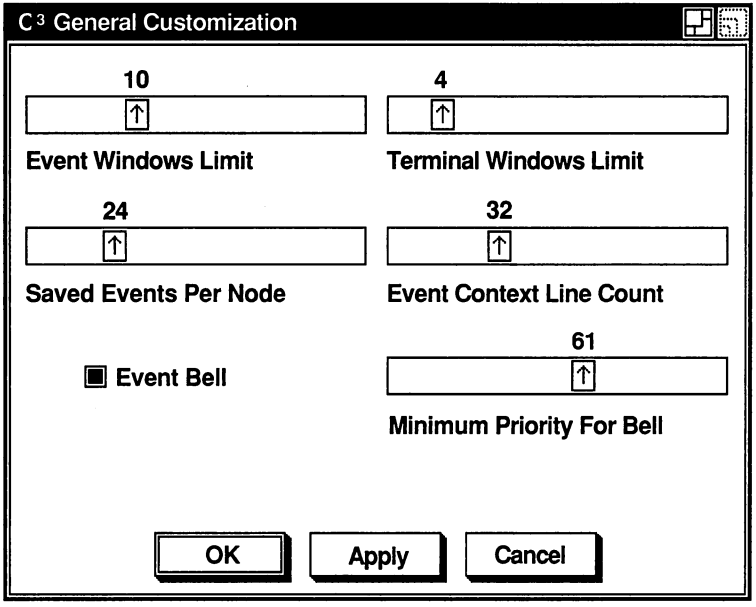


Log in at the **Username:** prompt. When you log out, the window closes.

2.9 Customizing C³ General Features



Choose the General . . . menu item from the Customize menu to change general features of the C³ display.



To change any of the values in the C³ General Customization dialog box, drag the arrow in the slider right or left, or move the pointer to a location on the slider and click MB1. Shaded buttons indicate the current settings.

Click on	To
OK	Apply your changes and close the dialog box.
Apply	Apply your changes without closing the dialog box.
Cancel	Close the dialog box without applying any changes.

From the C³ General Customization dialog box, you can modify any of the following:

Event Windows Limit

Limiting the number of event windows you can open conserves your host system resources.

Terminal Windows Limit

Limiting the number of terminal windows you can open conserves your host system resources. The C³ does not allow you to set this value larger than your process quotas.

Saved Events Per Node

Set the number of events you want to be able to view. The C³ will save the same number of events for each serviced node.

Event Context Line Count

Displaying fewer lines of event context in the event window increases the speed of the display.

Event Bell

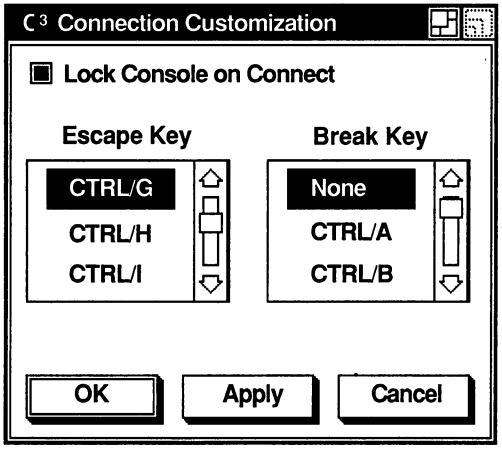
The button is shaded when the terminal bell is set to sound on the occurrence of events. Click on the Event Bell button or label to change its setting.

Minimum Priority for Bell

By default, all high priority (61–100) events cause the terminal bell to sound.

2.10 Customizing the Connect Interface

Choose the Connect . . . menu item from the Customize menu to change features of the Connect Interface.



From the C³ Connection Customization dialog box, you can enter any of the following:

Lock Console on Connect

The button is shaded when you have the exclusive input rights to a connection to a node's console line. When the node's console is locked, other users cannot send data to the node, but they can still examine log information and view your current activity. The node remains locked until you disconnect from the console by pressing the escape character.

Not locking the node means that you share input rights with other users on the specified console line. Any other operator is able to enter data simultaneously on the console line. Click on the Lock Console on Connect button or label to change its setting.

Escape Key

The escape key is a control key on your keyboard that you use to terminate the connection to the node's console. The individual Esc key cannot be used as the escape key for the Connect Interface. Valid escape keys are displayed in the list box.

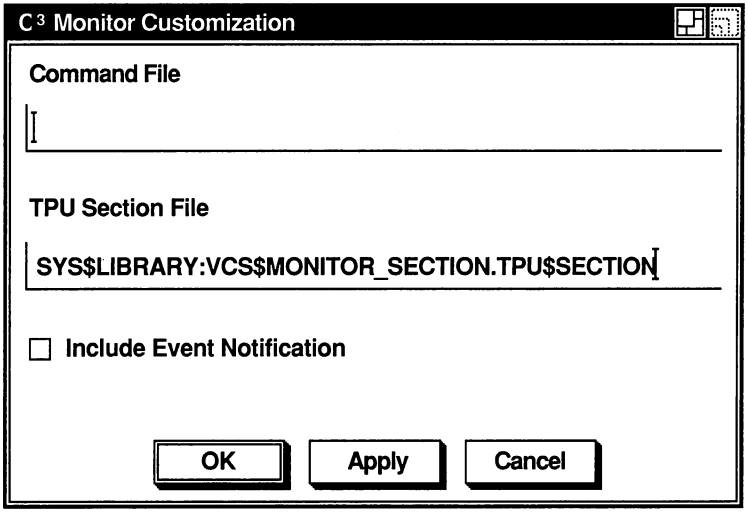
Break Key

The break key is a control key on your keyboard that you use to send a Break signal to the node to which you are connected. The individual Break key cannot be used as the break key for the Connect Interface. Valid break keys are displayed in the list box.

If you choose None, there is no way to send a Break signal to the connected node.

2.11 Customizing the Monitor Interface

Choose the Monitor . . . menu item from the Customize menu to change features of the Monitor Interface.



From the C³ Monitor Customization dialog box, you can enter any of the following:

Command File

Enter the name of a command file that contains Monitor Interface commands. Use a text editor to create this ASCII file. Enter the filename here to execute the commands in the file at the start of a Monitor Interface session.

If you do not enter a command filename here, no command file is executed.

TPU Section File

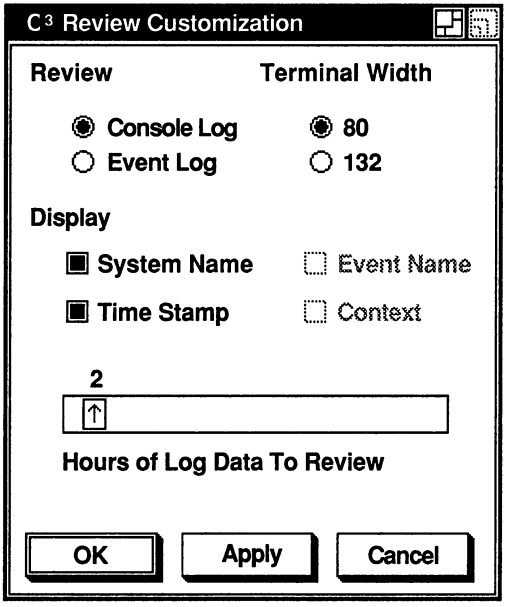
The TPU section file contains your custom key definitions to be used during the Monitor Interface session. Use the Monitor Interface DEFINE/KEY and SAVE commands to create key definitions and save them in a file. Then enter the section filename here to execute the file when you start a Monitor Interface session.

Include Event Notification

The button is shaded when the event summary window is set to be displayed on the screen as part of the Monitor Interface display. If you do not include event notification, the event indicators are not displayed on the screen when an event occurs and you cannot execute the Monitor Interface VIEW/EVENT command. The log window takes the two lines that otherwise would be used by the event summary window. Click on the Include Event Notification button or label to change its setting.

2.12 Customizing the Review Interface

Choose the Review . . . menu item from the Customize menu to change features of the Review Interface.



From the C³ Review Customization dialog box, you can enter any of the following:

Console Log

Click on the Console Log button or label to set the Review Interface to review console data.

Event Log

Click on the Event Log button or label to set the Review Interface to review event data. Use the Event Name and Context buttons to customize the Event Log review.

Terminal Width

Click on the 80 button if you want your review window to be 80 characters wide. Click on the 132 button if you want your review window to be 132 characters wide.

System Name

Click on the System Name button or label to set the Review Interface to display the name of the serviced node with each line of output data. Not including the system name is useful when reviewing data for a single node.

Time Stamp

Click on the Time Stamp button or label to set the Review Interface to display the time stamp with each line of output data.

Context

When the Event Log button is set, click on the Context button or label to display the context lines for each event in the event log output.

Event Name

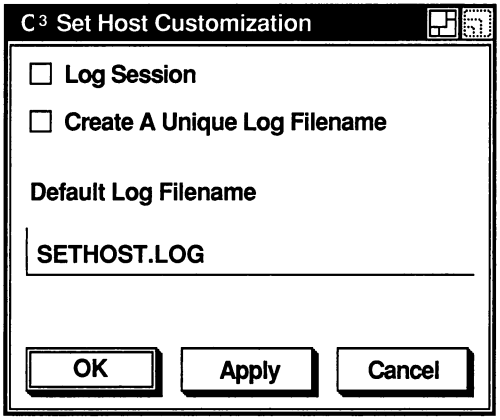
When the Event Log button is set, click on the Event Name button or label to display the event name, as defined in the configuration file, in the event line of the event log output.

Hours of Log Data to Review

By default, the time period for which log data is reviewed is from two hours ago until now. To change the number of hours, drag the arrow in the slider right or left, or move the pointer to a location on the slider and click MB1.

2.13 Customizing the Set Host Command

Choose the Set Host . . . menu item from the Customize menu to change features of the Set Host command.



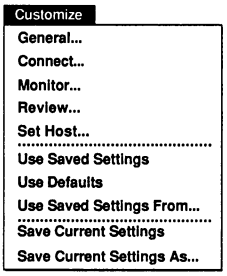
From the C3 Set Host Customization dialog box, you can enter any of the following:

Log Session
Click on the Log Session button to keep a log file of the entire session.

Create a Unique Log Filename
Click on the Create a Unique Log Filename button for the C3 to create a unique name for the log file. Otherwise, the default log filename is used.

Default Log Filename
Enter the log file name.

2.14 Saving and Restoring Customization Settings

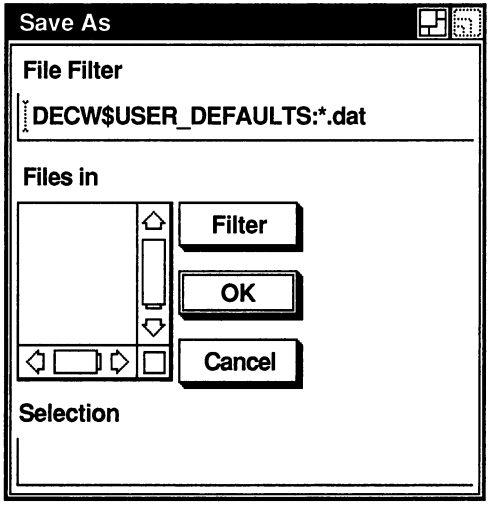


Once you modify C³ command settings, you can save your changes. When you save your changes, the C³ uses these settings each time you start the C³ using the VCS COORDINATE command.

2.14.1 Saving Customization Settings

To save customized settings:

- Choose the Save Current Settings menu item from the Customize menu for routine saving of customized commands.
- Choose the Save Current Settings As . . . menu item when you want to save the current customized commands with a new or different file name. The C³ displays a dialog box.



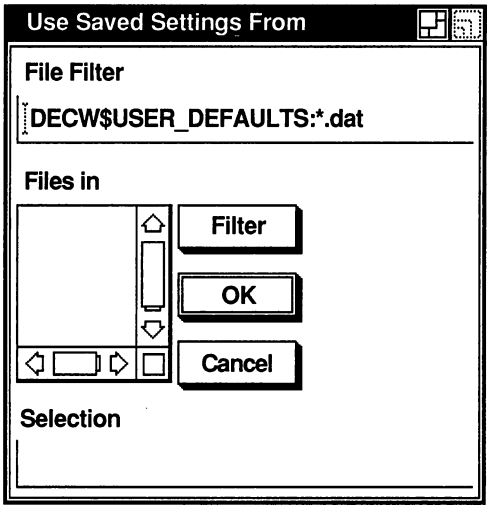
The dialog box prompts you for the name of the file you want to use to save your customized features. The file list box displays the names of any files in the current directory that contain customized features. By default, these files have the type DAT.

To use a file name that is not listed, enter the name of that file in the Selection text field and click on the OK button.

2.14.2 Restoring Customization Settings

To restore customized settings:

- Choose the Use Last Saved Settings menu item from the Customize menu to restore an existing file.
- Choose the Use Saved Settings From . . . menu item from the Customize menu to restore previously saved settings. The C³ displays a dialog box.



The dialog box prompts you for the name of the file you want to use to save your customized features. The file list box displays the names of any files in the current directory that contain customized features. By default, these files have the type DAT. Double click on the file name of the file you want to use.

Or, use the File Filter text field to list a subset of the files you can then select from. For example, to display all files in another directory with the file type DAT, type the complete directory specification in the File Filter text field and click on the Filter button. Then double click on the name of the file you want to use.

2.14.3 Customizing C³ DECterm Windows

You can save and restore the features of your C³ DECterm windows using the DECterm Customize menu and the VCS logical name VCS$TERMINAL_DEFAULT.

By default, the VCS logical name VCS$TERMINAL_DEFAULT is defined as SYS$LOGIN:DECW$TERMINAL_DEFAULT.DAT, which means that your C³ DECterm windows have the same characteristics as your other DECterm windows.

To customize all your C³ DECterm windows:

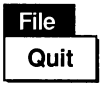
- Modify the DECterm window settings using the DECterm Customize menu.
- Save the current DECterm settings to a file.
- Define VCS$TERMINAL_DEFAULT to point to that file.

For example, save the DECterm settings you want for all your C³ windows to the file SYS$LOGIN:VCS$TERMINAL_DEFAULT.DAT. Then enter the following command:

```
$ DEFINE VCS$TERMINAL_DEFAULT SYS$LOGIN:VCS$TERMINAL_DEFAULT.DAT Return
```

Now all your C³ DECterm windows will have the features you saved.

2.15 Exiting from the C³



Choose Quit from the File menu to exit from the C³ and close the C³ window.

3

Creating the VCS Configuration with the Configuration Editor DECwindows Interface

The VCS configuration is composed of information describing its environment and the different entities it services. The VCS configuration information is static, contained in a single configuration file, and has to be updated manually. The Configuration Editor assists you in this task.

The Configuration Editor provides you with two interactive interfaces: an interface for the DECwindows environment and a command line interface for character-cell terminals.

This chapter describes the DECwindows interface. The character-cell command line interface is described in Chapter 4.

The Configuration Editor DECwindows interface enables you to assemble icons in the Editor window to represent the hardware layout of a computer room. The resulting window layout is used by the Central Control Coordinator (C^3), the principal operator interface to VCS.

3.1 Configuration File

The Configuration Editor is the first component you must use after the installation of the VCS product on your system, and before you can activate the product. You use the Configuration Editor to describe the environment in which you expect to operate, the entities to manage, and how to manage them.

These descriptions are saved in the VCS configuration file VCS$CONFIGURATION.DAT which is located in the directory SYS$MANAGER. The contents of the configuration file can be updated over time. Each time the managing components (such as the I/O Data Logger and Data Scanner) are activated, either interactively or during system startup, the latest version of these descriptions are implemented. You may update the configuration file while VCS is operational. However, the updated configuration information does not take effect until you enter the RECONFIGURE command.

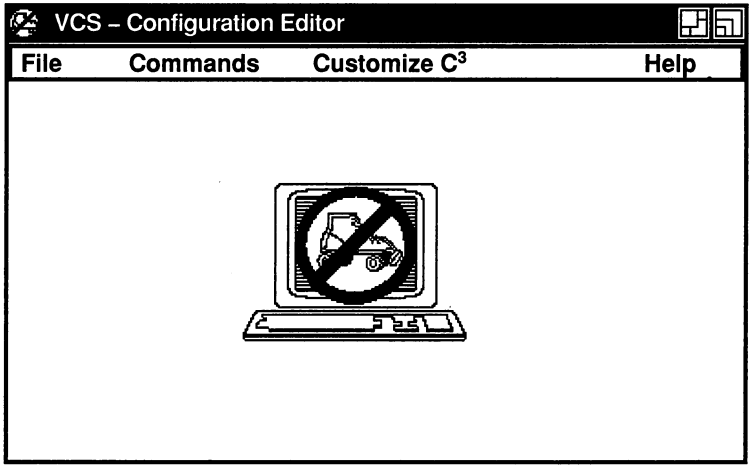
To avoid accidental activation of an incomplete VCS configuration at system startup (possibly due to a system reboot before you have completed editing the standard configuration file), the Configuration Editor allows you to redirect the new configuration information to a file that is different from the standard configuration file. The Configuration Editor also allows you to resume editing of a partially complete configuration file that is different from the standard configuration file. (See the /OUTPUT and /INPUT qualifiers for the DCL command VCS EDIT CONFIGURATION in Chapter 5.)

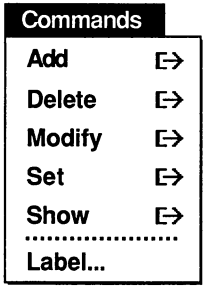
When you first install the VCS product, a template configuration file is created in the directory SYS$MANAGER. If a copy of the configuration file already exists, the VCS installation procedure does not create the template configuration file. This template configuration file contains the prerequisite information and a number of commonly defined information records which you may use to build your local site configuration information. If you prefer to start with a new configuration file, you may delete this template configuration file before invoking the Configuration Editor.

3.2 Starting the Configuration Editor

Note *You must have the SYSPRV privilege to edit the standard configuration file.*

To start the Configuration Editor, enter the command VCS EDIT CONFIGURATION at the DCL prompt. For complete details about the VCS EDIT CONFIGURATION command, see Chapter 5.

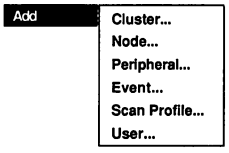




The Configuration Editor displays a window containing the VCS icon. The VCS icon represents the VCS host system. As you add nodes and peripherals to the configuration, their icons appear in the window.

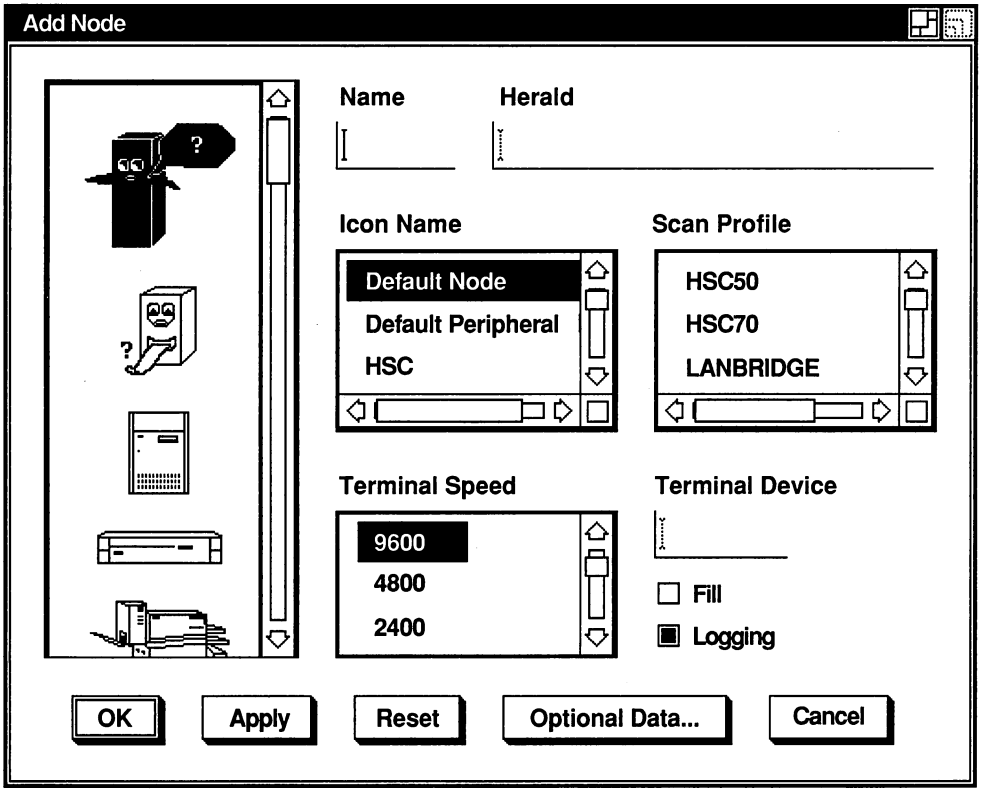
To add nodes, peripherals, and other information to the configuration, pull down the Commands menu and choose the Add submenu.

3.3 Adding a Node



The Add Node command enables you to enter the definition of a serviced node into the configuration. (If you need to define multiple nodes with very similar information, you can duplicate node definitions using the Modify submenu, as described in Section 3.11.1.)

To add a node to the configuration, choose the Node... menu item from the Add submenu. The Configuration Editor displays the Add Node dialog box.



Shaded buttons indicate the current settings.

Click on	To
OK	Apply your changes and close the dialog box.
Apply	Apply your changes without closing the dialog box.
Reset	Restore the last settings that you applied.
Optional Data...	Enter optional information about the serviced node.
Cancel	Close the dialog box without applying any changes.

When you apply your changes, the Configuration Editor displays a message that a new icon has been added to the editor window. The new node icon is highlighted in the center of the window to catch your attention. Drag the new icon to the position where you want it to appear in the C^3 window.

From the Add Node dialog box, you can enter any of the following:

Name
Enter the serviced node's name. The DECnet node name is recommended. The name must be from 1 to 12 characters in length. The Name and Terminal Device fields must not be left blank.

Herald
Enter the text string you want displayed in the Monitor Interface log window banner. The herald can be up to 63 characters in length.

Icon Name
Choose the icon representing the node. You can choose the icon by name from the list box or click on the icon itself in the icon box.

Scan Profile
Choose the scan profile for the serviced node from the list box. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the Configuration Editor displays a warning message if you apply your changes without choosing a scan profile.

Terminal Speed
Choose the speed of communications between the VCS host port and the serviced node's console port. By default, 9600 baud is selected.

Terminal Device

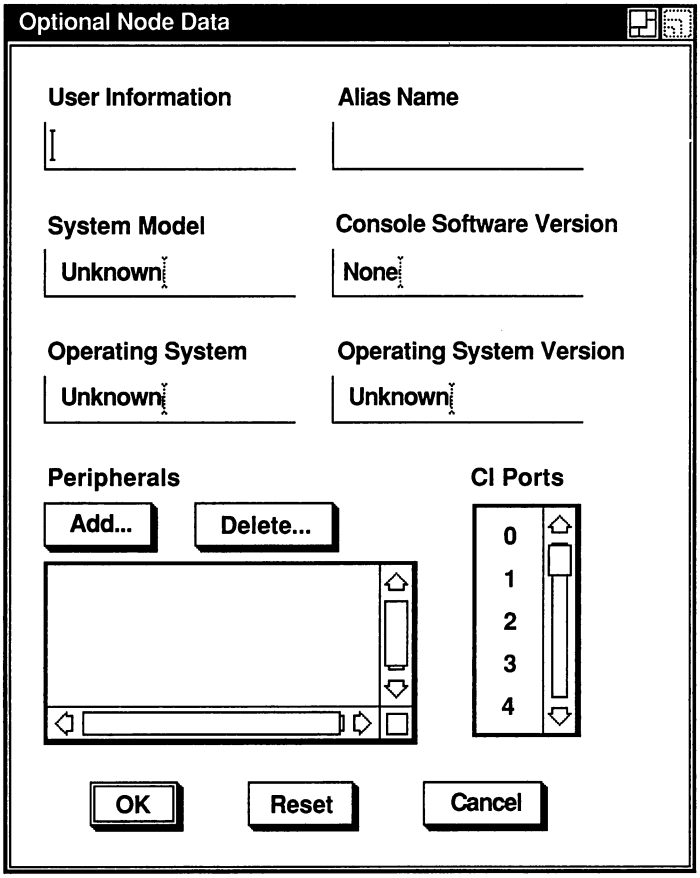
Enter the device name of the VCS host port that is connected to the serviced node's console port. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the Configuration Editor will not accept a blank Terminal Device field.

Fill

By default, data streams transmitted from the VCS software are not filled with a trailing null character. For HSC50 and HSC70 serviced nodes, choose Fill. Click on the button to change the setting.

Logging

By default, console data from the serviced node is logged by the VCS software. Click on the button to change the setting.



The Optional Node Data dialog box is a continuation of the Add Node dialog box. Your changes in the Optional Node Data dialog box are applied when you click on OK or Apply in the Add Node dialog box.

Click on	To
OK	Close the Optional Node Data dialog box.
Reset	Restore the last settings that you applied (in the Add Node dialog box).
Cancel	Ignore any changes and close the Optional Node Data dialog box.

From the Optional Node Data dialog box, you can enter any of the following:

User Information
Enter site-specific information, usually about operations. The user information can be up to 255 characters in length.

Alias Name
Enter the serviced system's DECnet node name if the DECnet node name is different from the node name you entered in the Add Node dialog box.

System Model
Enter the serviced system's model. The model can be up to 63 characters in length.

Console Software Version
Enter the serviced node's console software or firmware version. The version can be up to 63 characters in length.

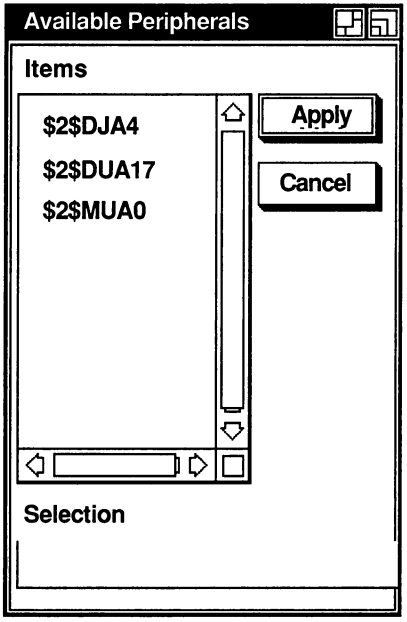
Operating System
Enter the serviced node's operating system or firmware. The operating system name can be up to 63 characters in length.

Operating System Version
Enter the serviced system's operating system version or firmware version. The version can be up to 63 characters in length.

Peripherals
Enter a list of peripheral devices that are connected to the serviced node. This list of peripheral devices is for reference. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the Configuration Editor only

accepts the names of peripheral devices that have been defined with the Add Peripheral command.

To add peripherals to the node definition, click on the Add... button. The Available Peripherals selection box displays a list of peripherals defined in the configuration.



Click on the name of the peripheral you want to add, then click on Apply.

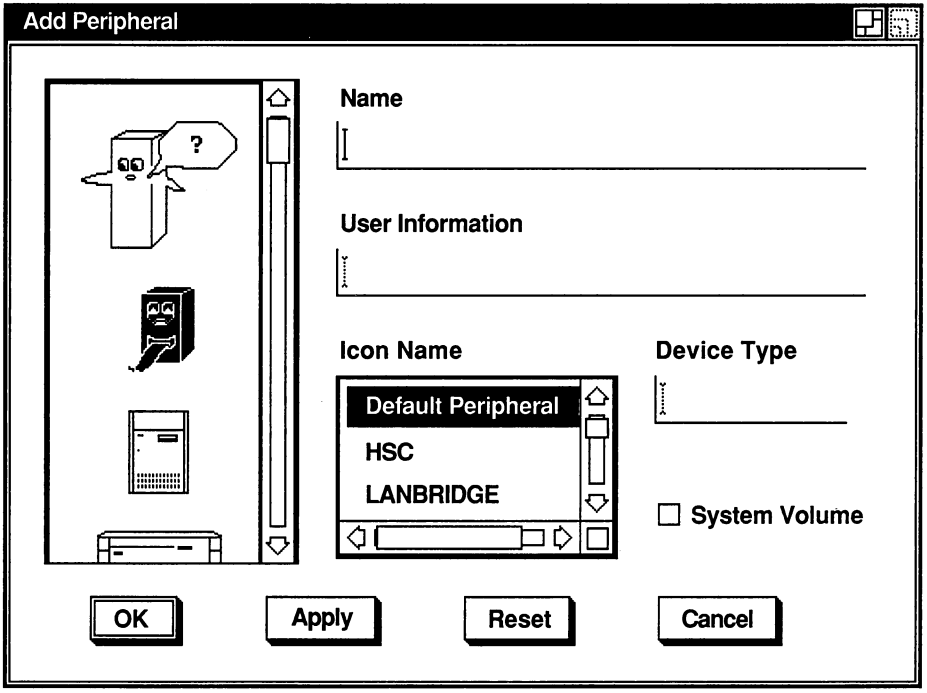
To remove a peripheral device from the node definition, click on the Delete... button. The Deletable Peripherals selection box displays a list of peripherals connected to the serviced node. Click on the name of the peripheral you want to delete, then click on Apply.

CI Ports

Choose each CI star coupler port to which the serviced node is connected. Click on a port number to choose a single port. Shift click on each port number to choose multiple ports. Shift click on one chosen port to remove it without removing other ports.

3.4 Adding a Peripheral

The Add Peripheral command enables you to enter into the configuration a definition of a peripheral device connected to a serviced node.



When you apply your changes, the Configuration Editor displays a message that a new icon has been added to the editor window. The new peripheral icon is highlighted in the center of the window to catch your attention. Drag the new icon to the position where you want it to appear in the C^3 window.

From the Add Peripheral dialog box, you can enter any of the following:

Name
Enter the name of the peripheral. The name must be from 1 to 63 characters in length.

User Information
Enter site-specific information. This field could be used to describe what the device is used for. The information can be up to 255 characters in length.

Icon Name

Choose the icon representing the peripheral. You can choose the icon by name from the list box or click on the icon itself in the icon box.

Device Type

Enter the peripheral device type. The device type can be up to 63 characters in length.

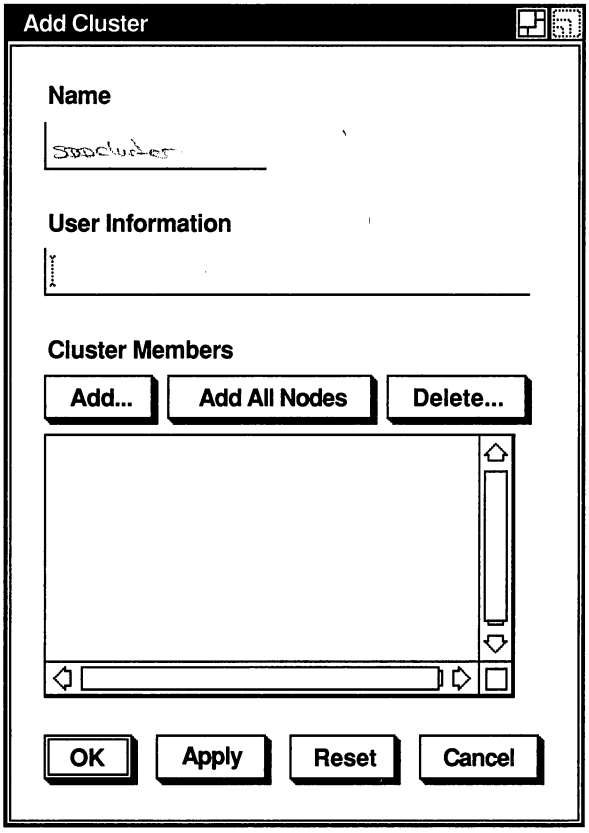
System Volume

When the System Volume button is shaded, it indicates that the peripheral device contains a system dependent volume. Click on the button to change the setting.

This button is used as a reminder that the operation of a system or cluster depends on the presence of this volume.

3.5 Adding a Cluster

The Add Cluster command enables you to define a list of serviced nodes that compose the membership of a VAXcluster. This list allows you to refer to the serviced nodes as a group.



From the Add Cluster dialog box, you can enter any of the following:

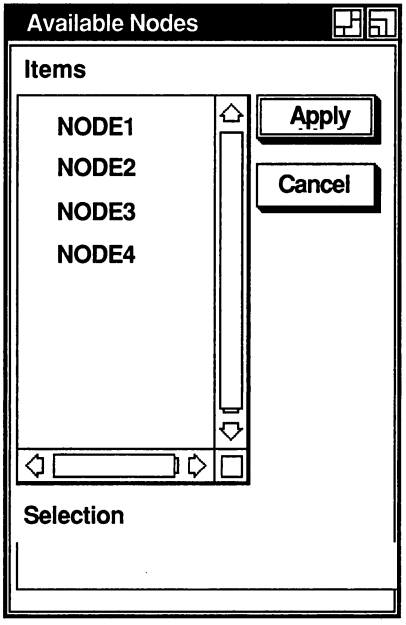
Name
Enter the name of the VAXcluster. The name must be from 1 to 12 characters in length.

User Information
Enter a text string of up to 255 characters containing site-specific information. This field could be used to describe the operations of the VAXcluster.

Cluster Members

Enter a list of nodes composing the VAXcluster. Use the Add Node command to create the definition for each node in the configuration. The maximum number of nodes you can include on this list is 32.

To add nodes to the Cluster Members list, click on the Add... button. The Available Nodes selection box displays a list of nodes defined in the configuration.



Click on the name of the node you want to add, then click on Apply. The new name is displayed in the Cluster Members list.

To delete nodes from the Cluster Members list, click on the Delete... button. The Deletable Nodes selection box displays a list of nodes in the Cluster Members list. Click on the name of the node you want to delete, then click on Apply.

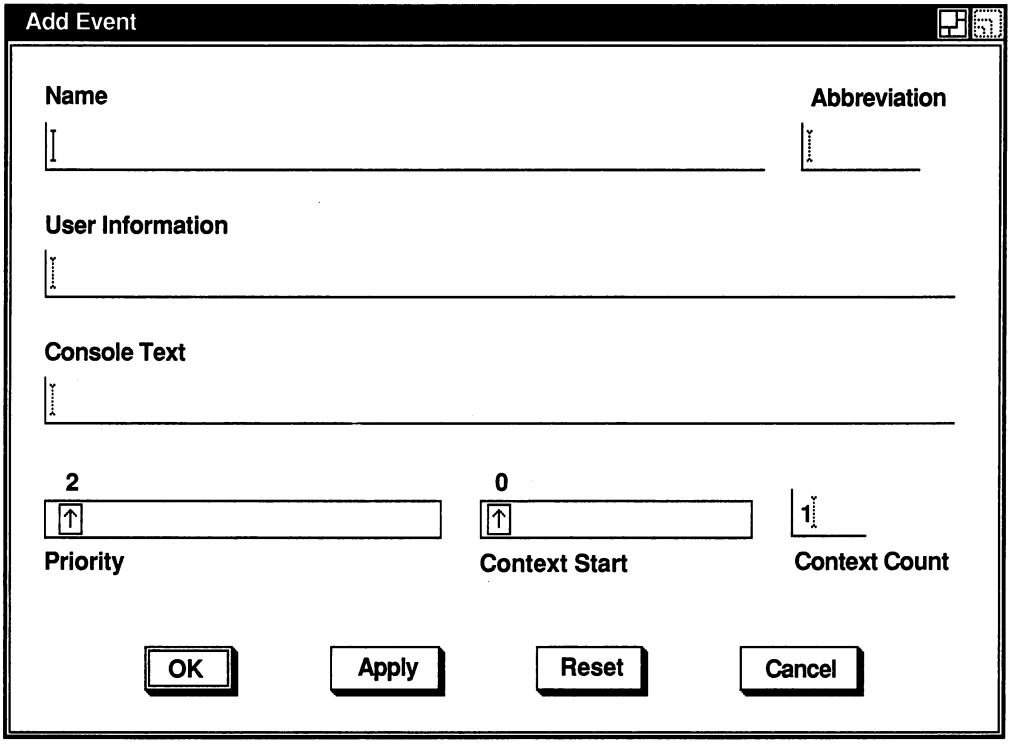
Click on the Add All Nodes button to indicate that all existing nodes in the configuration are the members of this VAXcluster. All nodes in the configuration are displayed in the list box.

3.6 Adding an Event

The Add Event dialog box enables you to create an instruction directing the Data Scanner on:

- How to detect a console message displayed by the serviced nodes.
- What circumstantial information to retain once the specified console message is recognized.
- What additional information, together with the above, to relate to the Access Interface user applications.

Use the Add Scan Profile command to assemble these instructions into scan profiles.



From the Add Event dialog box, you can enter any of the following:

Name
Enter the name of the event. The name must be from 1 to 63 characters in length.

Abbreviation
Enter an abbreviated name of the event. The abbreviated name must be from 1 to 6 characters in length. This abbreviated name is used by the Monitor Interface for display in its Event Summary window.

User Information
Enter a text string of up to 255 characters. The Data Scanner passes this text string unchanged to the Access Interface. The Access Interface makes the information available directly to the user application code. This field could be used to direct Access Interface user applications on how to respond to a particular event.

Console Text
Enter a text string of up to 132 characters. This text string (usually a system message) must match a text string generated by a serviced node. The Data Scanner scans for a contiguous text string. You cannot enter wild-card characters or multiple lines. Therefore, the text you enter must be identical to a contiguous sequence of characters in a node message.

Priority
To change this number, drag the arrow in the slider right or left, or move the pointer to a location on the slider and click MB1.

Priority is a decimal number that indicates the relative importance of an event. The higher the number, the more important the event. You can assign numbers from 2 to 98. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), you must enter a text string in the Console Text field when you specify events with priority 2 to 98.

VCS reserves the numbers 0, 1, 99, and 100. If you use these numbers, you are, in effect, renaming these reserved events. Unless you turn data verification off, events with priority 0, 1, 99, and 100 must have a blank Console Text field.

Context Start

To change this number, drag the arrow in the slider right or left, or move the pointer to a location on the slider and click MB1.

Context Start is the number of console text lines preceding the event. This value is a decimal number ranging from 0 to 63. A context start of **n** indicates the **n**-th preceding console text line whose time stamp is to be returned.

The Data Scanner processes this value to provide the time stamp of the context information. This time stamp will be used as the starting point of console log file retrieval. The command VCS REVIEW/EVENT=CONTEXT shows how this field affects log file data retrieval. The Access Interface makes the time stamp available directly to the user application code.

The Data Scanner returns the context start time stamp identical to the one of the event message.

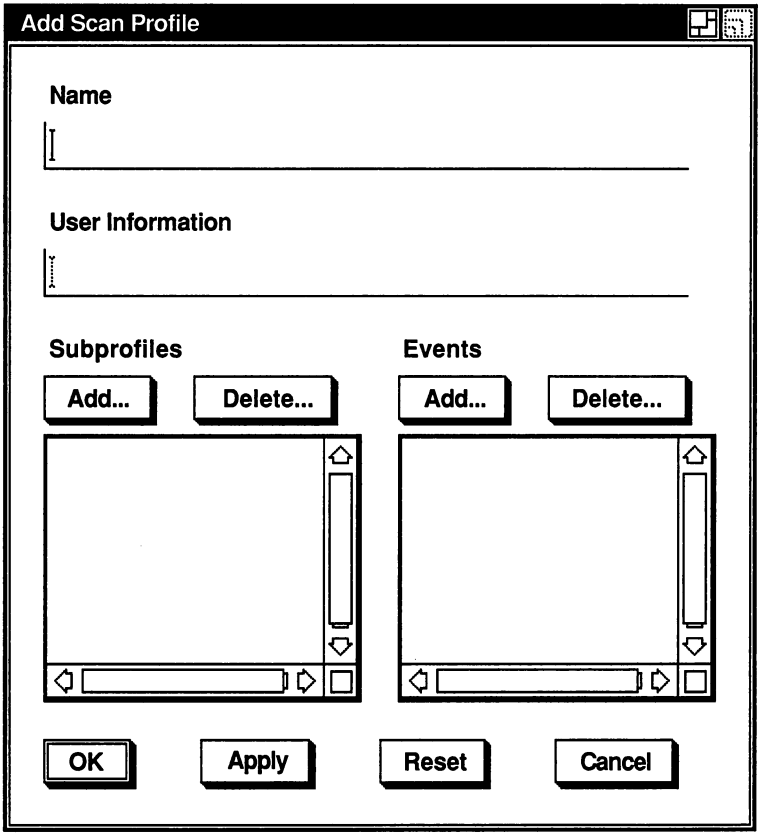
Context Count

Enter the total number of context lines. This value is a decimal number ranging from 0 to 1023. A context count of **n** indicates **n**-number of console text lines that make up the context data of the event, beginning with the one identified with Context Start.

The Monitor Interface uses the information to determine how many lines of console data to display when an event context is viewed. The Access Interface makes the information available directly to the user application code.

3.7 Adding a Scan Profile

The Add Scan Profile command enables you to add into the VCS configuration the definition of a scan profile. A scan profile is a collection of events that instructs the Data Scanner how to look for console messages in the console data stream from a serviced node.



From the Add Scan Profile dialog box, you can enter any of the following:

Name
Enter the name of the scan profile. The name must be from 1 to 63 characters in length.

User Information

Enter a text string of up to 255 characters containing site-specific information. This field could be used to contain a description of the scan profile.

Subprofiles

Enter a list of other scan profiles to make up this scan profile. A maximum number of 34 combined scan profiles and events may make up this scan profile. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the Configuration Editor will not accept recursive or undefined scan profiles.

To add scan profiles, click on the Add... button. The Available Profiles selection box displays a list of scan profiles defined in the configuration. Click on the name of the scan profile you want to add, then click on Apply. The new name is displayed in the Subprofiles list.

To remove a scan profile, click on the Delete... button. The Deletable Profiles selection box displays a list of scan profiles in the Subprofiles list. Click on the name of the profile you want to delete, then click on Apply.

Events

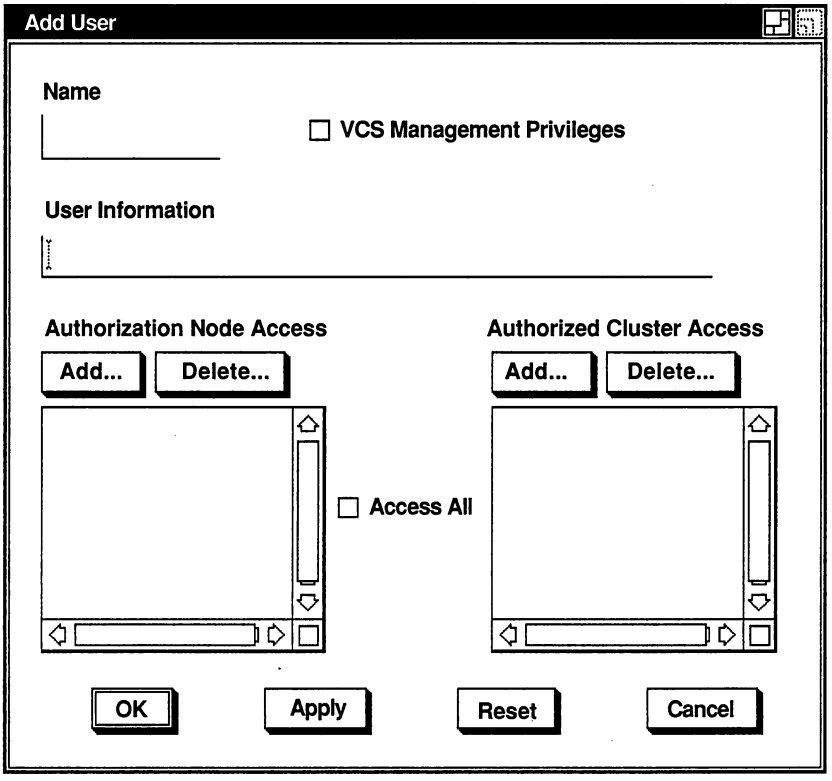
Enter a list of events that make up this scan profile. Use the Add Event command to define an event in the configuration. A maximum number of 34 combined events and subprofiles may make up this scan profile. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the Configuration Editor will not accept undefined events.

To add events to the scan profile, click on the Add... button. The Available Events selection box displays a list of events defined in the configuration. Click on the name of the event you want to add, then click on Apply. The new name is displayed in the Events list.

To remove an event, click on the Delete... button. The Deletable Events selection box displays a list of events in the Events list. Click on the name of the event you want to delete, then click on Apply.

3.8 Adding a VCS User

The Add User command enables you to define a VCS user and that user's access to serviced nodes and VCS commands. By default, access is not restricted. To restrict access, use the Authenticate Users For Security button described in Section 3.9.



From the Add User dialog box, you can enter any of the following:

Name
Enter the name of an authorized user. The name must be from 1 to 12 characters in length.

VCS Management Privileges
By default, if access is restricted, a defined user is not authorized to perform the VCS management functions: DISABLE, ENABLE, RECONFIGURE, and SHUTDOWN. Click on the button or label to change this setting.

User Information

Enter a text string of up to 255 characters containing site-specific information. This field could be used to contain a description of the user's operations duties and organizational affiliates.

Authorized Node Access

By default, if access is restricted, the user is authorized to access only selected individual nodes. The VCS node access commands are CONNECT, MONITOR, OUTPUT, REVIEW, and SELECT.

To add a node to the Authorized Node Access list, click on the Add... button. The Available Nodes selection box displays a list of nodes in the configuration. Click on the name of the node you want to add, then click on Apply. The new node is displayed in the Authorized Node Access list.

To remove a node from the user's access, click on the Delete... button. The Deletable Nodes selection box displays a list of nodes in the Authorized Node Access list. Click on the name of the node you want to delete, then click on Apply.

Authorized Cluster Access

By default, if access is restricted, the user is authorized to access only selected VAXclusters. The VCS access commands are CONNECT, MONITOR, OUTPUT, REVIEW, and SELECT.

To add a cluster to the Authorized Cluster Access list, click on the Add... button. The Available Clusters selection box displays a list of clusters in the configuration. Click on the name of the cluster you want to add, then click on Apply. The new node is displayed in the Authorized Node Access list.

To remove a cluster, click on the Delete... button. The Deletable Clusters selection box displays a list of clusters in the Authorized Cluster Access list. Click on the name of the cluster you want to delete, then click on Apply.

Access All

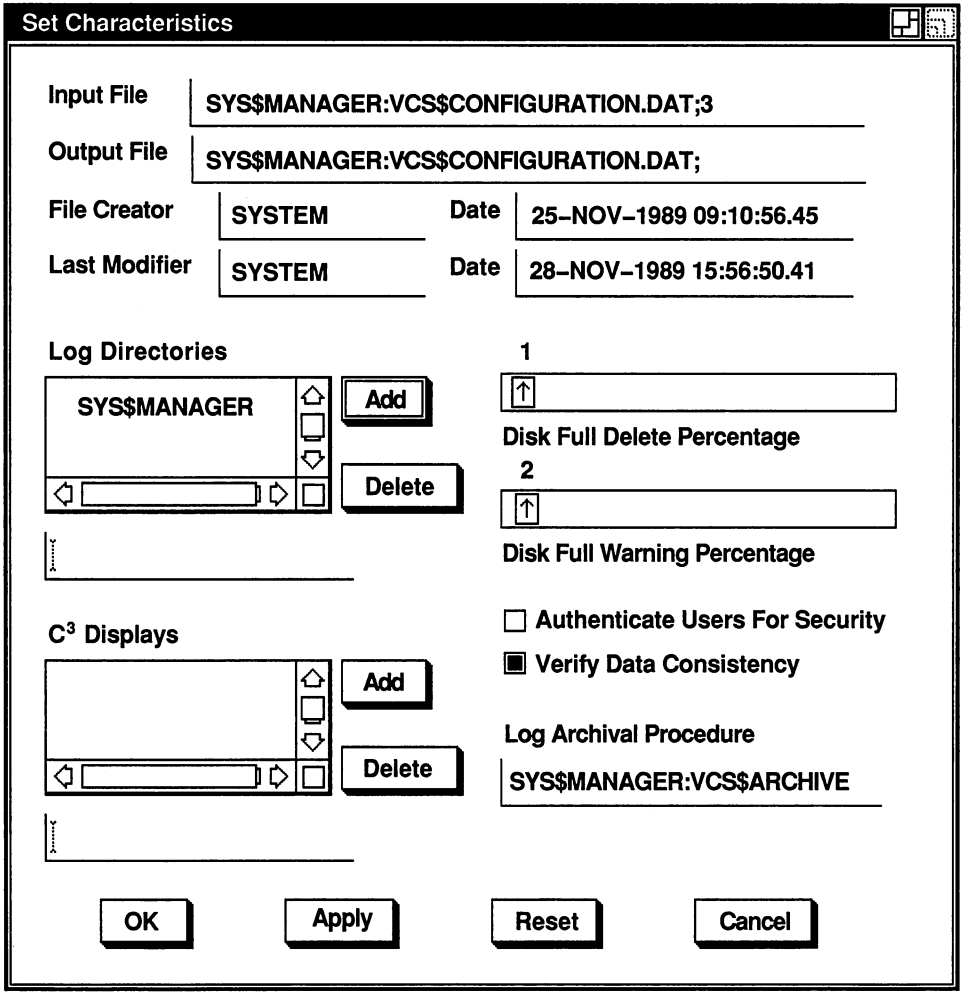
Select Access All to authorize user access to all serviced nodes. The Add... and Delete... buttons are dimmed.

3.9 Setting Up the VCS Environment

To define the characteristics of the VCS operating environment, pull down the Commands menu and choose the Characteristics... menu item from the Set submenu.

To display the characteristics of the VCS environment, you can also click MB2 on the VCS host icon in the Configuration Editor window and choose Show... from the pop-up menu.

The Set Characteristics command defines the characteristics of the VCS operating environment.



From the Set Characteristics dialog box, you can enter any of the following:

Log Directories

This list of disk directories forms the VCS log directory search list. The new log files are created in the first directory of the search list. The older log files can reside in any directory of the search list. Unless you turn data verification off (see the Verify Data Consistency button), the log directories must exist on the VCS host system.

To insert a new directory into the list, click on one of the existing directories. The selected existing directory name is highlighted. Enter the new directory name in the text field. When you click on the Add button, the new directory is placed above the selected directory. If you do not select a position, the new directory is placed at the end of the list.

To delete a directory from the list, click on the directory name, then click on the Delete button.

C^3 Displays

The C^3 Displays list allows you to define the DECwindows display locations for Central Control Coordinator interfaces. Enter each DECwindows display in the following format, *node::server.screen*.

where:

node::	specifies the DECwindows workstation, accessible over a DECwindows transport mechanism, on which you want to display the C^3 interface.
server.screen	specifies the DECwindows display server and display monitor.

The DECwindows transport mechanism (for example, DECnet) is used to send input from the user to the C^3 process and output from the C^3 process to the workstation display.

You can specify a maximum of eight different displays. The default C^3 display is 0::0.0 which is the VCS host system.

When you start VCS on the host system, VCS starts a C^3 interface for each display listed in the C^3 Displays list.

Log Archival Procedure

Enter the file name of a DCL procedure. If you do not specify a file type, the Editor uses the default file type of COM. No wildcard characters are allowed in the file specification. The default procedure is SYS$MANAGER:VCS$ARCHIVE.COM.

This DCL procedure is executed when the disk is full at the level specified by the Disk Full Warning Percentage. If no archiving procedure is specified, when the Disk Full Delete Percentage is reached, VCS deletes the oldest log files in the current log directory to accomodate the newly received data.

Disk Full Delete Percentage

This setting pertains to the disk device where the VCS log directory is located. When unused space on the disk device falls below this percentage, VCS deletes the oldest console and event log files.

To change this percentage, drag the arrow in the slider right or left, or move the pointer to a location on the slider and click MB1. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the disk full warning percentage must be less than the disk full delete percentage.

Disk Full Warning Percentage

This setting pertains to the disk device where the VCS log directory is located. When unused space on the disk device falls below this percentage, VCS issues a warning. If a Log Archival Procedure has been specified, it is executed.

To change this percentage, drag the arrow in the slider right or left, or move the pointer to a location on the slider and click MB1. Unless you turn data verification off (see the Verify Data Consistency button in Section 3.9), the disk full warning percentage must be greater than the disk full delete percentage.

Authenticate Users For Security

By default, the VCS security facility is inactive. Click on the button to change the setting.

The VCS security facility controls user access to VCS and its serviced entities. You use the Add User submenu to define this access control. The Authenticate Users For Security button allows you to activate these access restrictions. Deactivating the VCS security facility leaves the access control definitions intact.

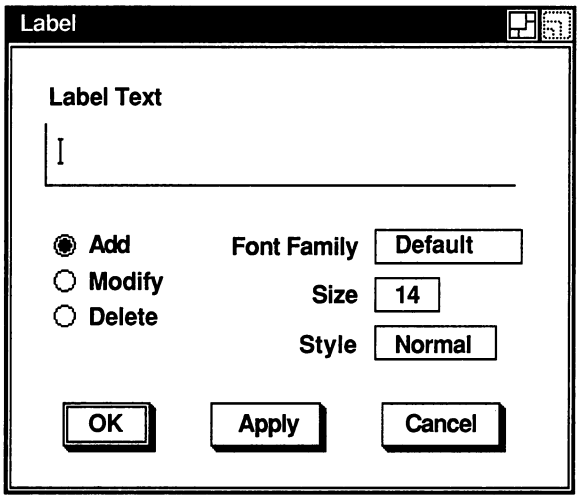
Verify Data Consistency

By default, the VCS data verification facility is turned on. Click on the button to change the setting.

Turning off data verification allows you to run the Configuration Editor to create a configuration file for a different VCS host system.

3.10 Labels

To create a label for the Configuration Editor window, pull down the Commands menu and choose the Label... menu item. The Configuration Editor displays a dialog box.



When you apply your changes, the Configuration Editor displays a message that a new label has been added to the editor window. The new label is highlighted in the center of the window to catch your attention. Drag the new label to the position where you want it to appear in the C^3 window.

From the Label dialog box, you can enter any of the following:

Label Text

Enter the text of the label you want to add, modify, or delete. The Label Text field displays your label in the font family, size, and style you have chosen. This allows you to see what your label will look like before you add it to or modify it in the editor window. The label can be up to 63 characters long. Leading and trailing spaces are removed.

Add

Click on the Add button to create a new label when you next apply your changes. Enter the label text you want to add. The Label Text field accepts upper and lower case.

Modify

Click on the Modify button to change an existing label when you next apply your changes. Enter the label text you want to modify. The Label Text field is sensitive to upper and lower case.

Delete

Click on the Delete button to delete a label. Enter the label text you want to delete. The Label Text field is sensitive to upper and lower case.

Font Family

Click on the Font Family box or label to display a pop-up menu of fonts.

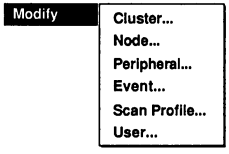
Size

Click on the Size box or label to display a pop-up menu of font sizes. Size menu items are dimmed when you choose the default Font Family.

Style

Click on the Style box or label to display a pop-up menu of font styles. Style menu items are dimmed when you choose the default Font Family.

3.11 Modifying the Configuration

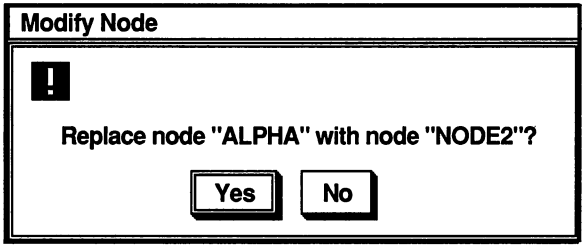


To modify nodes, peripherals, and other definitions in the configuration, pull down the Commands menu and choose the Modify submenu. You can also click MB2 on an icon in the Configuration Editor window and choose Modify from the pop-up menu.

To modify any definition in the VCS configuration, choose the menu item you want to change from the Modify submenu. For example, to modify a node definition, choose the Node menu item. The Modify Node selection box displays a list of nodes defined in the configuration. Click on the node you want to modify and click on the Apply button. The Configuration Editor displays a Modify Node dialog box which has the same contents as the Add Node dialog box described in Section 3.3. Enter your changes in the Modify Node dialog box just as you did in the Add Node dialog box.

3.11.1 Adding Duplicate Nodes

To duplicate node information, enter a different node name and a different terminal device in the Modify Node dialog box. When you apply the changes, you are prompted with a message like the following one:



If you respond by clicking on No, the Configuration Editor adds a new node to the configuration. In this example, a response of No adds node NODE2 to the configuration. The new node information is the same as that of the first node, except for the name and terminal device. If you respond with Yes, only the node name and terminal device are changed.

3.11.2 Deleting from the Configuration

To delete any definition in the configuration, choose the menu item from the Delete submenu. For example, to delete a node from the configuration, choose the Node menu item. The Configuration Editor displays a Delete Node selection box. Click on the name of the node you want to delete, then click on Apply.

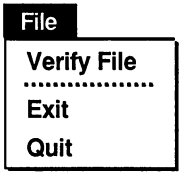
You can also click MB2 on an icon in the Configuration Editor window and choose Delete from the pop-up menu.

3.12 Customizing the Central Control Coordinator

The Configuration Editor allows you to customize the C^3. For details about customizing C^3 commands, see Chapter 2.

If you customize the C^3 commands from the Configuration Editor, you must save C^3 customization settings using the menu items described in Section 2.14. Exiting from the Configuration Editor does not save C^3 customization settings.

3.13 Verifying the Configuration



Choose the Verify File menu item to verify the consistency of all the configuration file information.

An example of inconsistency is a peripheral device entered as part of a node definition, but not defined itself with the Add Peripheral command. Verification also fails if all icons are not visible in the window. When verification is turned on, the consistency of individual definitions in the configuration is checked whenever the definitions are added or modified.

If the configuration file is found to have inconsistent information, the Configuration Editor displays a dialog box that lists the inconsistencies. You can then correct the inconsistencies and reverify the file. VCS cannot run with an unsuccessfully verified configuration file.

3.14 Exiting from the Configuration Editor

To end an editing session without saving your changes to the configuration file, choose the Quit menu item from the File menu.

To save your changes and end the editing session, choose the Exit menu item from the File menu.

The Configuration Editor verifies the configuration file for consistency whenever you exit an editing session.

If the verification fails with warnings, you are prompted whether you want to save the unsuccessfully verified file or resume editing to correct the inconsistencies.

If the verification fails with errors, you are prompted whether you want to save the unsuccessfully verified file as a journal file or whether you want to resume editing to correct the inconsistencies.

You must turn data verification off (see the Verify Data Consistency button in Section 3.9) to save unsuccessfully verified configuration data to the standard configuration file.

4

Configuration Editor Commands

This chapter is a reference to the Configuration Editor command iine interface for use with character-cell terminals. The Configuration Editor commands described in this chapter are listed in Table 4–1.

Table 4–1 Configuration Editor Commands

@	EXIT	QUIT
ADD CLUSTER	HELP	SET DISPLAY
ADD EVENT	MODIFY CLUSTER	SET LOGGING
ADD NODE	MODIFY EVENT	SET SECURITY
ADD PERIPHERAL	MODIFY NODE	SET VERIFY
ADD SCAN	MODIFY PERIPHERAL	SHOW
ADD USER	MODIFY SCAN	SPAWN
DELETE	MODIFY USER	VERIFY

@ (Execute Command File)

Executes an Editor command file.

Format

@ *file-name*

Parameters

file-name
is the Editor command file to be executed. If you do not specify a file type, the Editor uses the default file type of COM. No wildcard characters are allowed in the file specification.

Description

When you specify the @ command to execute a command file at the **VCS EDIT>** prompt, the Editor switches the input command stream source from the terminal to the specified file. The file should contain commands as they would be typed at the **VCS EDIT>** prompt. Commands that need arguments must contain the arguments in the command line.

Example

```
$ VCS EDIT CONFIGURATION Return
VCS EDIT> @VCS$CONFIGURATION_TEMPLATE.COM Return
```

This example shows a command file, named VCS$CONFIGURATION_TEMPLATE.COM, that creates a template configuration file. The command file is then invoked within the Editor. A copy of the file VCS$CONFIGURATION_TEMPLATE.COM is located in the VCS$EXAMPLES: area.

ADD

Adds a new entity to the VCS configuration.

Format

ADD *entity entity-name (/qualifier . . .)*

Parameters

entity
is a keyword denoting an entity in the VCS configuration. The valid entities are:

- CLUSTER
- EVENT
- NODE
- PERIPHERAL
- SCAN
- USER

entity-name
is the name of the specified entity to be added to the VCS configuration.

Description

Use the ADD command to create a new entry in the VCS configuration.

ADD CLUSTER

Adds a new entry to the VCS configuration describing the node membership of a VAXcluster.

Format

ADD CLUSTER *cluster-name (/qualifier . . .)*

Parameters

cluster-name
is the name of the VAXcluster. The name must be from 1 to 12 characters in length.

Qualifiers

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the VAXcluster operations nature.

The default site specific information is a null string.

/MEMBER=(node-name(, . . .))
is a list of nodes composing the VAXcluster. Use the ADD NODE command to create the entry for a node in the configuration. It is possible to include the name of a node that is not currently in the configuration, however, you will be cautioned for doing so. The maximum number of nodes you can include on this list is 32.

You may use the wildcard character, asterisk (*), to indicate all existing nodes in the configuration to be the members of this VAXcluster.

Description

The ADD CLUSTER command allows you to create a list of serviced nodes that make up the membership of a VAXcluster of the same name. This list of names would allow you to conveniently reference the serviced nodes.

Example

```
VCS EDIT> ADD CLUSTER GROUP1 - Return
_VCS EDIT> /INFORMATION= - Return
_VCS EDIT> "HPS Software Engineering Development VAXcluster" - Return
_VCS EDIT> /MEMBER=(ALPHA,BETA,GAMMA,CHI,PSI) Return
VCS EDIT>
```

This example shows the creation of an entry describing the VAXcluster GROUP1 which is comprised of three VAXs (ALPHA, BETA, and GAMMA) and two HSCs (CHI and PSI). These nodes must be previously defined using the ADD NODE command. You can change the membership by using the MODIFY CLUSTER command.

ADD EVENT

Adds a new entry to the VCS configuration describing an event to be triggered when a message is detected on the console of a serviced node.

Format

ADD EVENT *event-name (/qualifier . . .)*

Parameters

event-name
is the name of the event. The name must be from 1 to 63 characters in length.

Qualifiers

/ABBREVIATION=abbreviated-event-name
is an abbreviated name of the event. The abbreviated name must be from 1 to 6 characters in length. This abbreviated name is used by the Monitor Interface for display in its Event Summary window.

The default abbreviation is the first up to 6 characters from the event name.

/COUNT=context-count
defines the total number of context lines. This value is a decimal number ranging from 0 to 1023. A context count of **n** indicates **n**-number of console text lines that make up the context data of the event, beginning with the one identified with /START.

The Monitor Interface uses the information to determine how many lines of console data to display when an event context is viewed. The Access Interface makes the information available directly to the user application code.

The default context count is 1.

/INFORMATION=site-specific-directive
is a quoted text string of up to 255 characters in length. The Data Scanner passes this text string unchanged to the Access Interface. The Access Interface makes the information available directly to the user application code. This field may be used for encoding instructions directing the various Access Interface user applications how to response to a particular event.

The default directive is a null string.

/PRIORITY=priority-number
is a decimal number that indicates the relative importance of an event. You can assign numbers from 2 to 98. The higher the number, the more important the event. VCS reserves the numbers 0, 1, 99, and 100.

The default priority is 2.

/START=context-start
is the beginning of the context for this event. This value is a decimal number ranging from 0 to 63. A context start of **n** indicates the **n**-th preceding console text line whose time stamp is to be returned.

The Data Scanner processes this value to provide the time stamp of the context information. This time stamp will be used as the starting point of console log file retrieval. The Monitor Interface command REVIEW/EVENT=CONTEXT shows how this field affects log file data retrieval. The Access Interface makes the time stamp available directly to the user application code.

The default context start is 0. (The Data Scanner returns the context start time stamp identical to the one of the event message.)

/TEXT=message-text
is a quoted text string of up to 132 characters in length. This text string (usually a system message) must match a text string generated by a serviced node. The Data Scanner scans for a contiguous text string. Therefore, the text you use must be identical to a contiguous sequence of characters in a node message.

ADD EVENT

Description

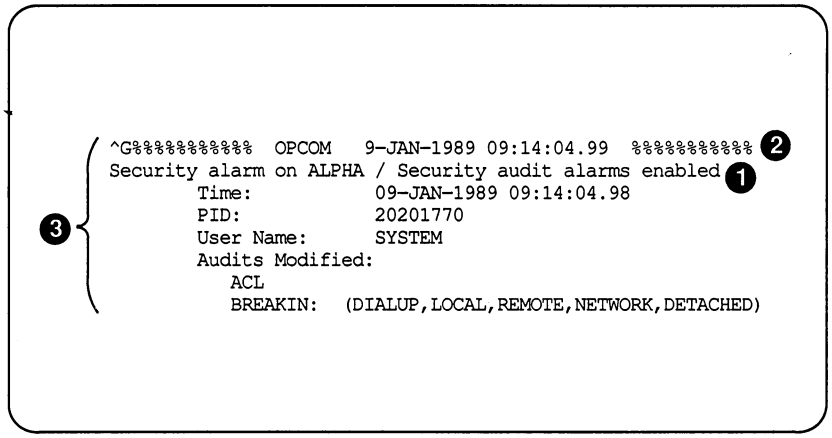
A console event is usually considered as the receipt of a console message from a serviced node. The console message may span several text lines. A specific portion of a console message, that can be uniquely identified by the Data Scanner, may fall anywhere within these text lines. An event's context is defined as a number of contiguous console text lines. Thus, to fully recognize an *event*, the identifiable text portion has to be placed in *context*.

The ADD EVENT command allows you to create an *instruction* directing the Data Scanner on:

- How to detect a console message displayed by the serviced nodes.
- What circumstantial information to retain once the specified console message is recognized.
- What additional information, together with the above, to relate to the Access Interface user applications.

These instructions can be assembled into various scan profiles using the ADD SCAN command. The *anatomy* of a console message, as seen by the Data Scanner, is described in Figure 4–1.

Figure 4–1 Event Anatomy



MR–4082–RA

❶ The text string to be searched for in the console data stream by the Data Scanner. Use the /TEXT qualifier to specify this text string.

You can be as specific as you want in specifying this text string. If you specify the entire message **Security alarm on ALPHA / Security audit alarms enabled**, only such messages (the ones that come from node ALPHA) are recognized by the Data Scanner. However, if you specify only a static portion of the message, for example **Security alarm on** or **Security audit alarms enabled**, the Data Scanner would recognize all messages that have such string in their text bodies (that is, those of the same type but coming from all nodes serviced by VCS).

The text string must be a contiguous sequence of characters as appears in the actual message body. The text string cannot span more than one line of console text.

❷ The text string alone may trigger a false event if it is recognized out of context. Use the /START qualifier to direct the Data Scanner to also pass along the time stamp of the preceeding text line. This time stamp can then be used as the starting point to retrieve the entire message from the console log file.

❸ Finally, a complete message may consist of several text lines. Use the /COUNT qualifier to indicate that fact. This value can then be used in conjunction with the context start time stamp in analyzing the message context.

If verification is enabled, the following items are validated:

- Events of priority 2 to 98 must have console text specified.
- Events of priority 0, 1, 99, and 100 (VCS reserved events) must have **no** console text specified.

Example

```
VCS EDIT> ADD EVENT SECURITY_AUDIT_ALARM - Return
_VCS EDIT> /ABBREVIATION="SecAud" /PRIORITY=60 - Return
_VCS EDIT> /TEXT= - Return
_VCS EDIT> "Security alarm on ALPHA / Security audit alarms enabled" - Return
_VCS EDIT> /START=1 /COUNT=10 Return
VCS EDIT>
```

This example shows how to create the instruction for scanning an OPCOM message similar to the above. If the following OPCOM message is received on a console line:

```
06:01:35.73 %%%%%%%%%%  OPCOM  30-NOV-1988 06:01:35.72  %%%%%%%%%%
06:01:35.74 Security alarm on ALPHA / Security audit alarms enabled
06:01:35.75     Time:           30-NOV-1988 06:01:35.72
06:01:35.76     PID:            20401A93
06:01:35.77     User Name:      SYSTEM
06:01:35.78     Audits Modified:
06:01:35.79         ACL
06:01:35.80         AUTHORIZATION
06:01:35.81         AUDIT
06:01:35.82         BREAKIN:    (DIALUP,LOCAL,REMOTE,NETWORK,DETACHED)
```

The following key information is made available by the Data Scanner:

- the event name **SECURITY_AUDIT_ALARM**
- the event name abbreviation **SecAud**
- the event priority **60**
- the event time stamp **06:01:35.74**
- the context start time stamp **06:01:35.73**
- the context count **10**

Refer to Chapter 9 for details of all information that the user Access Interface application code can expect from the Data Scanner.

ADD NODE

Adds a new entry to the VCS configuration describing a serviced node.

Format

ADD NODE *node-name (/qualifier . . .)*

Parameters

node-name
is the name that you assign to a serviced node in the VCS configuration. The name must be from 1 to 12 characters in length. It is recommended that this name be the same as the serviced system's DECnet node name where it is applicable.

Qualifiers

/ALIAS=DECnet-node-name
is the serviced system's DECnet node name if it is different from the specified node name. This name must be identical to that in the VCS host system's DECnet database.

The default DECnet node name is the specified node name.

/CONSOLE=version
is a quoted text string of up to 63 characters in length denoting the serviced node's console software or firmware version.

The default console version is "NONE".

/FILL
/NOFILL (D)
indicates how a data stream is to be sent by VCS to the serviced node over the console physical connection. Data streams can be transmitted from VCS using the Monitor Interface OUTPUT command or the Connect Interface CONNECT/INPUT command.

If /FILL is specified, each character in the data stream is padded with a trailing null character (000), before the entire stream is transmitted to the serviced node. This is done to prevent VCS from overrunning the serviced node console device.

It is recommended that you specify /FILL for the HSC50 and HSC70 nodes.

The default is /NOFILL.

/HERALD=node-identification
is a quoted text string of up to 63 characters in length containing a site-specific description of the node for display purposes.

The default node identification is a null string.

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the node operations nature.

The default site specific information is a null string.

/LOGGING (D)
/NOLOGGING
indicates if the console data from the serviced node is to be logged by VCS. This characteristic can also be dynamically changed using the Monitor Interface DISABLE/LOGGING and ENABLE/LOGGING commands.

The default is /LOGGING.

/MODEL=system-model
is a quoted text string of up to 63 characters in length denoting the serviced node's system model.

The default hardware model is "UNKNOWN".

/PERIPHERAL=(peripheral-name(, . . .))
is a list of the peripheral devices that are connected to the serviced node. Use the ADD PERIPHERAL command to create the entry for a peripheral device in the configuration.

The default is none.

/PORT=(ci-port-number(, . . .))
is a list of the CI ports that connect the serviced node to a CI star coupler.

The default is none.

/SCAN=scan-name
is the scan profile which is a collection of instructions that directs the Data Scanner how to detect predetermined console messages from the serviced node. Use the ADD SCAN command to create the entry for a scan profile in the configuration.

The default is none.

/SPEED=baud-rate
is the baud rate of the EIA physical link between the VCS host system's serial line interface port and the service node's console port.

The default speed is 9600 baud.

/SYSTEM=operating-system
is a quoted text string of up to 63 characters in length denoting the serviced node's operating system or firmware.

The default operating system name is "UNKNOWN".

/TERMINAL=device-name
is the device name of the VCS host system's EIA serial line interface port which is linked to the serviced node's console port.

/VERSION=version
is a quoted text string of up to 63 characters in length denoting the serviced node's operating system or firmware version.

The default operating system version is "UNKNOWN".

Description

The ADD NODE command allows you to create the description of a node serviced by VCS. Each node is managed by VCS as a generic entity. That is, VCS does not impose any contextual interpretation of the console data received from a serviced node at the *node* level. At the node level, VCS treats the console data as plain ASCII data streams.

For convenience, the nodes may be grouped and identified as a VAXcluster (ADD CLUSTER command).

When you specify the /CONSOLE or /VERSION qualifiers, Digital recommends the following standard format for the version number as described in the *Digital VAX-11 Software Engineering Manual.* A version number consists of a compound string constructed of the following discrete items:

```
support version . update - edit patch
```

where:

support identifies the support level of the program with a single capital letter (or null). The following are support levels:

B	Benchmark version
D	Demonstration version
S	Special customer version
T	Field Test version
V	Released or frozen version
X	Unsupported or experimental version

version identifies the major release, generation, or base level of a program.

update indicates a minor release with (if present) a period followed by a single decimal digit.

edit identifies any alteration of the source code with (if present) a minus sign followed by a maintenance number.

patch (if present) is a single capital letter identifying an alteration of the program's binary object form.

If verification is enabled, the following items are validated:

- The specified peripheral devices must exist in the configuration database.
- The terminal device must exist on the VCS host system.
- The Editor displays a warning message if no scan profile is specified.

Example

```
VCS EDIT>  ADD NODE ALPHA - Return
_VCS EDIT>  /HERALD="HPS Software Engineering Development System" - Return
_VCS EDIT>  /INFORMATION= - Return
_VCS EDIT>  "For scheduling stand-alone time, please see Mr. Luthor." - Return
_VCS EDIT>  /MODEL="VAX 8800" /CONSOLE="8.1" - Return
_VCS EDIT>  /SYSTEM="VMS" /VERSION="5.0-2" - Return
_VCS EDIT>  /PERIPHERAL=($2$MUA0,$2$MUA1) - Return
_VCS EDIT>  /PORT=(5,15) /SCAN=VMS8800 - Return
_VCS EDIT>  /TERMINAL=LTA30 /SPEED=9600  Return
VCS EDIT>
```

This example shows the description of node ALPHA.

ADD PERIPHERAL

Adds a new entry to the VCS configuration describing a peripheral device connected to a serviced node.

Format

ADD PERIPHERAL *peripheral-name (/qualifier . . .)*

Parameters

peripheral-name
is the name of the peripheral. The name must be from 1 to 63 characters in length.

Qualifiers

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the device operations nature.

The default site specific information is a null string.

/SYSTEM_VOLUME
/NOSYSTEM_VOLUME (D)
indicates whether the peripheral device contains a system dependent volume such that removing it, or the system it is attached to, from a logically formed collection of systems (for example, a VAXcluster) renders that group of systems, as a whole, inoperational.

The default is /NOSYSTEM_VOLUME.

/TYPE=peripheral-type
is a quoted text string of up to 63 characters in length denoting the peripheral device type.

The default peripheral device type is "UNKNOWN".

ADD PERIPHERAL

Description

The ADD PERIPHERAL command allows you to create the description of a peripheral device that is connected to one or more serviced nodes.

Example

```
VCS EDIT> ADD PERIPHERAL $2$MUA0 /TYPE="TA79" - Return
_VCS EDIT> /INFORMATION="Tape drive for daily backup only." Return
VCS EDIT>
```

This example shows the description of the TA79 tape drive 2MUA0.

ADD SCAN

Adds a new entry to the VCS configuration describing a scan profile.

Format

ADD SCAN *scan-name (/qualifier . . .)*

Parameters

scan-name
is the name of the scan profile. The name must be from 1 to 63 characters in length.

Qualifiers

/EVENT=(event-name(, . . .))
is a list of events. Use the ADD EVENT command to create the entry for an event in the configuration. A maximum number of 34 events and scan profiles combined may be included on this list.

The default is none.

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the scan profile.

The default site specific information is a null string.

/SCAN=(scan-name(, . . .))
is a list of other scan profiles to make up this scan profile. Recursive scan profile inclusion is not allowed. A maximum number of 34 scan profiles and events may be included on this list.

The default is none.

ADD SCAN

Description

The ADD SCAN command allows you to assemble a collection of up to 128 events to instruct the Data Scanner how to look for predetermined console messages in the console data stream from a serviced node. The events can be included individually with the /EVENT qualifier or collectively with the /SCAN qualifier. Either /EVENT or /SCAN qualifier must be specified in the command.

Both qualifiers may be specified in the command. When both are specified, the events, in the order they are specified, are included first in the combined list of events. The scan profiles, in the order they are specified, are included next in the combined list of events. These scan profiles are not evaluated by the Editor. That is, the events contained in these scan profiles are not broken down and included directly in the combined list of events. This evaluation is to be done only when the newly created scan profile is processed by the Data Scanner.

If verification is enabled, the following items are validated:

- The specified events must exist in the configuration database.
- The specified scan profiles must exist in the configuration database.
- Recursive scan profile inclusion is not allowed.

The following guidelines can help you create efficient and accurate scan profiles:

- Minimize the number of events to scan. Although you can supply a maximum of 128 events for each node, the fewer events you supply, the faster the scanning process.
- Supply as many contiguous characters as you can to describe the event. VCS scans most efficiently for longer events. Fewer than seven characters may cause performance problems during scanning. Moreover, longer strings are less likely to raise false alarms. Very short strings sometimes appear in console messages you prefer to ignore.
- Include non-printable characters in the event if they are part of the message.

- Specify events in the scan profile by descending priority. Although this sequencing of events is not required, it will help you to maintain the scan profiles.
- You can assign a single priority number to one or more events. However, if you place two events of equal priority in the scan profile, VCS searches until it finds the first occurrence of the event and matches it with the text string it is searching for. VCS does not continue to the second event of equal priority. For each line of text received from a node, VCS scans for the text string associated with the highest priority event first, and the text string associated with the lowest priority event last.

  This principle also applies to the order in which you specify the scan profiles that make up another scan profile. If you have two events of the same priority located in two different profiles for a node, then VCS detects the event in the first profile.

Example

```
VCS EDIT> ADD SCAN ALPHA - Return
_VCS EDIT> /INFORMATION= - Return
_VCS EDIT> "HPS Software Engineering Development System" - Return
_VCS EDIT> /EVENT=(CAD_SOFTWARE_MESSAGES,CAD_PLOTTER_ERRORS) - Return
_VCS EDIT> /SCAN=(VAX_8840,VMS_VERSION_5) Return
VCS EDIT>
```

This example shows the scan profile ALPHA that is intended for the serviced node ALPHA, which is a VAX 8840, running VMS Version 5.0 and supporting CAD activities. This scan profile directs the Data Scanner to look for selected VAX 8840 hardware error messages, selected VMS Version 5.0 system error messages, and those informational messages that are issued by the CAD software and the error messages from a connected CAD plotter device.

ADD USER

ADD USER

Adds a new entry to the VCS configuration describing an authorized user's access privileges.

Format

ADD USER *user-name (/qualifier . . .)*

Parameters

user-name
is the name of an authorized user. The name must be from 1 to 12 characters in length.

Qualifiers

/AUTHORIZED_ACCESS=({ ***node-name(, . . .)*** / ***cluster-name(, . . .)*** } ***)***
is a list containing the names of the individual serviced nodes, or those represented by a defined VAXcluster name, which the user is authorized to access. See Table 4–2 for the list of access commands. Use the ADD NODE command to create the entry for a serviced node in the configuration. Use the ADD CLUSTER command to create the list of node members of a VAXcluster.

You may use the wildcard character, asterisk (*), to indicate all serviced nodes to be authorized for access by the user.

The default is none.

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the user's operations duties and organization affiliates.

The default site specific information is a null string.

/MANAGEMENT
/NOMANAGEMENT (D)
indicates if the user is authorized to perform VCS management functions. See Table 4–3 for the list of management commands.

The default is /NOMANAGEMENT.

Description

The ADD USER command allows you to register a user name and describe the access restrictions imposed upon a user. These restrictions are applicable only when security is enabled with the SET SECURITY command. By default, security is not enabled. Table 4–2 contains the VCS commands the user may use to access the serviced nodes' console devices and recorded console data. Table 4–3 contains the VCS commands the user may use to alter the VCS operations status.

Table 4–2 Node Access Commands

Command	Interfaces	Function
CONNECT	Central Control Coordinator DCL Monitor	Accesses a serviced node's console device for read and write.
MONITOR	Central Control Coordinator DCL	Monitors serviced nodes' console devices.
OUTPUT	Monitor	Writes to multiple serviced nodes' console devices.
REVIEW	Central Control Coordinator DCL Monitor	Retrieves serviced node's historical console data.
SELECT	Monitor	Accesses a serviced node's console device for read and write.

Table 4–3 VCS Management Commands

Command	Interfaces	Function
DISABLE	Monitor	Disables service of a node's console device.
ENABLE	Monitor	Enables service of a node's console device.
RECONFIGURE	DCL Monitor	Activates the latest configuration information.
SHUTDOWN	DCL Monitor	Shuts down the VCS software.

Example

```
VCS EDIT>  ADD CLUSTER ENGVXC /MEMBER=(ALPHA,BETA,GAMMA,DELTA) Return
VCS EDIT>  ADD USER VOBA - Return
_VCS EDIT>  /INFORMATION="Third-Shift Engineering Lab Lead Operator" - Return
_VCS EDIT>  /NOMANAGEMENT /AUTHORIZED=(ENGVXC)  Return
VCS EDIT>
```

This example shows the user VOBA, who is the third-shift lead operator of the engineering computer lab, being authorized to access only the engineering VAXcluster ENGVXC, comprised of four systems: ALPHA, BETA, GAMMA, and DELTA.

DELETE

Deletes an existing entity in the VCS configuration.

Format

DELETE { **CLUSTER** *cluster-name* | **EVENT** *event-name* | **NODE** *node-name* | **PERIPHERAL** *peripheral-name* | **SCAN** *scan-name* | **USER** *user-name* }

Parameters

cluster-name
is the name of the VAXcluster to be deleted. You may use the wildcard character, asterisk (*), to indicate all existing VAXcluster entries to be deleted from the configuration.

event-name
is the name of the event to be deleted. You may use the wildcard character, asterisk (*), to indicate all existing event entries to be deleted from the configuration.

node-name
is the name that you assign to a serviced node in the VCS configuration to be deleted. You may use the wildcard character, asterisk (*), to indicate all existing nodes to be deleted from the configuration.

peripheral-name
is the name of the peripheral to be deleted. You may use the wildcard character, asterisk (*), to indicate all existing peripheral devices to be deleted from the configuration.

DELETE

scan-name
is the name of the scan profile to be deleted. You may use the wildcard character, asterisk (*), to indicate all existing scan profiles to be deleted from the configuration.

user-name
is the name of an authorized user to be deleted. You may use the wildcard character, asterisk (*), to indicate all existing authorized users to be deleted from the configuration.

Description

Use the DELETE command to delete an existing entity or the entire class of an entity from the configuration.

Examples

1
```
VCS EDIT> DELETE NODE ALPHA Return
VCS EDIT>
```

This command deletes the entry of node ALPHA from the configuration.

2
```
VCS EDIT> DELETE EVENT * Return
VCS EDIT>
```

This command deletes all event entries from the configuration.

EXIT

Terminates an editing session.

Format

EXIT

Description

The EXIT command terminates an editing session and returns to the DCL command level. You can also issue the EXIT command by pressing Ctrl/Z. To enter an editing session again, use the VCS EDIT CONFIGURATION command at the DCL command level.

A number of steps are taken before the Editor terminates successfully:

1 If verification is in effect, the configuration data is verified for consistency. If an inconsistency is found, it is identified and you are given the choice either to ignore the inconsistency and end the editing session, or to remain in the Editor to correct the inconsistency. If you ignore the inconsistency and end the editing session, you can resume editing the journal file at a later time by using the /INPUT qualifier.

2 If changes have been made to the configuration data, a new version of the VCS configuration file is created.

Example

```
VCS EDIT>  SET VERIFY  Return
VCS EDIT>  EXIT  Return
%VCS-E-SCANNOF, Scan VMS sub-scan OPER not found in configuration file.
%VCS-I-CONTEXIT, Changes will not be saved - Continue exit? [No]  Return
VCS EDIT>
```

This example shows an inconsistency found in the configuration data when the EXIT command is entered. The user chooses not to terminate the editing session and remains in the Editor.

HELP

Invokes the Configuration Editor online help facility.

Format

HELP *(topic)*

Parameters

topic
is the name of the command for which you requested help. If you omit the topic, the Editor displays a list of available topics.

Description

The help facility displays information about requested Editor topics. At the **VCS EDIT>** prompt, you can display a list of topics by typing HELP and pressing Return. The Editor responds by displaying a brief description of how to use the help facility, followed by a list of topics for which help is available. The prompt is changed to **Topic?**. In response to the **Topic?** prompt, you can:

- Enter one of the topic names from the list.
- Press Return one or more times to exit, or press Ctrl/Z to immediately exit from from the help facility.
- Enter a question mark (?) to see a list of topics.
- Enter an asterisk (*) to display all available information from the help facility.
- Enter a topic immediately followed by an ellipsis (. . .) to display all of the information on a topic. For example, ADD . . . displays all of the information available for the ADD command.

You can abbreviate any topic name, although ambiguous abbreviations result in all matches being displayed.

Example

```
VCS EDIT> HELP ADD Return
```

When you enter this command, the Editor prints a description about how to use the ADD command.

MODIFY

Edits the information of an existing entity in the VCS configuration.

Format

MODIFY *entity entity-name (/qualifier . . .)*

Parameters

entity
is a keyword denoting an entity in the VCS configuration. The valid entities are:

- CLUSTER
- EVENT
- NODE
- PERIPHERAL
- SCAN
- USER

entity-name
is the name of the specified entity in the VCS configuration to be modified.

Description

Use the MODIFY command to edit the information of an existing entity in the VCS configuration.

MODIFY CLUSTER

Edits the information of an existing entry in the VCS configuration describing the node membership of a VAXcluster.

Format

MODIFY CLUSTER *cluster-name (/qualifier . . .)*

Parameters

cluster-name
is the name of the VAXcluster.

Qualifiers

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the VAXcluster operations nature.

/MEMBER=(node-name(, . . .))
/NOMEMBER=(node-name(, . . .))
is a list of nodes to be added to or removed from the VAXcluster list of members. Use the ADD NODE command to create the entry for a node in the configuration.

Description

The MODIFY CLUSTER command allows you to edit the information of an existing entry describing a VAXcluster configuration. See the ADD CLUSTER command for more information.

Example

```
VCS EDIT> MODIFY CLUSTER GROUP1 /NOMEMBER=BETA Return
VCS EDIT>
```

This example shows the removal of node BETA from the GROUP1 VAXcluster configuration.

MODIFY EVENT

Edits the information of an existing entry in the VCS configuration describing an event to be triggered when a message is detected on the console of a serviced node.

Format

MODIFY EVENT *event-name (/qualifier . . .)*

Parameters

event-name
is the name of the event.

Qualifiers

/ABBREVIATION=abbreviated-event-name
is an abbreviated name of the event. The abbreviated name must be from 1 to 6 characters in length. This abbreviated name is used by the Monitor Interface for display in its Event Summary window.

/COUNT=context-count
defines the total number of context lines. This value is a decimal number ranging from 0 to 1023. A context count of **n** indicates **n**-number of console text lines that make up the context data of the event, beginning with the one identified with /START.

The Monitor Interface uses the information to determine how many lines of console data to display when an event context is viewed. The Access Interface makes the information available directly to the user application code.

/INFORMATION=site-specific-directive
is a quoted text string of up to 255 characters in length. The Data Scanner passes this text string unchanged to the Access Interface. The Access Interface makes the information available directly to the user application code. This field may be used for encoding instructions directing the various Access Interface user applications how to respond to a particular event.

/PRIORITY=priority-number
is a decimal number that indicates the relative importance of an event. You can assign numbers from 2 to 98. The higher the number, the more important the event. VCS reserves the numbers 0, 1, 99, and 100.

/START=context-start
is the beginning of the context for this event. This value is a decimal number ranging from 0 to 63. A context start of **n** indicates the **n**-th preceding console text line whose time stamp is to be returned.

The Data Scanner processes this value to provide the time stamp of the context information. This time stamp will be used as the starting point of console log file retrieval. The Monitor Interface command REVIEW/EVENT=CONTEXT shows how this field affects log file data retrieval. The Access Interface makes the time stamp available directly to the user application code.

/TEXT=message-text
is a quoted text string of up to 132 characters in length. This text string (usually a system message) must match a text string generated by a serviced node. The Data Scanner scans for a contiguous text string. Therefore, the text you use must be identical to a contiguous sequence of characters in a node message.

Description

The MODIFY EVENT command allows you to edit the information of an existing entry describing an event. See the ADD EVENT command for more information.

Example

```
VCS EDIT>  MODIFY EVENT SECURITY_AUDIT_ALARM /PRIORITY=30 Return
VCS EDIT>
```

This example shows how to change the priority of the event SECURITY_AUDIT_ALARM to 30.

MODIFY NODE

Edits the information of an existing entry in the VCS configuration describing a serviced node.

Format

MODIFY NODE *node-name (/qualifier . . .)*

Parameters

node-name
is the name that you assign to a serviced node in the VCS configuration.

Qualifiers

/ALIAS=DECnet-node-name
is the serviced system's DECnet node name if it is different from the specified node name. This name must be identical to that in the VCS host system's DECnet database.

/CONSOLE=version
is a quoted text string of up to 63 characters in length denoting the serviced node's console software or firmware version.

/FILL
/NOFILL
indicates how a data stream is to be sent by VCS to the serviced node over the console physical connection. Data streams can be transmitted from VCS using the Monitor Interface OUTPUT command or the Connect Interface CONNECT/INPUT command.

If /FILL is specified, each character in the data stream is padded with a trailing null character (000), before the entire stream is transmitted to the serviced node. This is done to prevent VCS from overrunning the serviced node console device.

It is recommended that you specify /FILL for the HSC50 and HSC70 nodes.

/HERALD=node-identification
is a quoted text string of up to 63 characters in length containing a site-specific description of the node for display purposes.

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the node operations nature.

/LOGGING
/NOLOGGING
indicates if the console data from the serviced node is to be logged by VCS. This characteristic can also be dynamically changed using the Monitor Interface DISABLE/LOGGING and ENABLE/LOGGING commands.

/MODEL=system-model
is a quoted text string of up to 63 characters in length denoting the serviced node's system model.

/PERIPHERAL=(peripheral-name(, . . .))
/NOPERIPHERAL=(peripheral-name(, . . .))
is a list of the peripheral devices to be added to or removed from the serviced node. Use the ADD PERIPHERAL command to create the entry for a peripheral device in the configuration.

/PORT=(ci-port-number(, . . .))
/NOPORT=(ci-port-number(, . . .))
is a list of the CI ports to be added to or removed from the list of the CI ports that connect the serviced node to a CI star coupler.

/SCAN=scan-name
is the scan profile which is a collection of instructions that directs the Data Scanner how to detect predetermined console messages from the serviced node. Use the ADD SCAN command to create the entry for a scan profile in the configuration.

/SPEED=baud-rate
is the baud rate of the EIA physical link between the VCS host system's serial line interface port and the service node's console port.

MODIFY NODE

/SYSTEM=operating-system
is a quoted text string of up to 63 characters in length denoting the serviced node's operating system or firmware.

/TERMINAL=device-name
is the device name of the VCS host system's EIA serial line interface port which is linked to the serviced node's console port.

/VERSION=version
is a quoted text string of up to 63 characters in length denoting the serviced node's operating system or firmware version.

Description

The MODIFY NODE command allows you to edit the information of an existing entry describing a node. See the ADD NODE command for more information.

Example

```
VCS EDIT>  MODIFY NODE ALPHA /TERMINAL=LTA30 /SPEED=9600 Return
VCS EDIT>
```

This example shows a new terminal device and its associated speed is being assigned to node ALPHA.

MODIFY PERIPHERAL

Edits the information of an existing entry in the VCS configuration describing a peripheral device connected to a serviced node.

Format

MODIFY PERIPHERAL *peripheral-name (/qualifier . . .)*

Parameters

peripheral-name
is the name of the peripheral.

Qualifiers

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the device operations nature.

/SYSTEM_VOLUME
/NOSYSTEM_VOLUME
indicates whether the peripheral device contains a system dependent volume such that removing it, or the system it is attached to, from a logically formed collection of systems (for example, a VAXcluster) renders that group of systems, as a whole, inoperational.

/TYPE=peripheral-type
is a quoted text string of up to 63 characters in length denoting the peripheral device type.

MODIFY PERIPHERAL

Description

The MODIFY PERIPERHAL command allows you to edit the information of an existing entry describing a peripheral device. See the ADD PERIPHERAL command for more information.

Example

```
VCS EDIT>  MODIFY PERIPHERAL $2$MUA0 /TYPE="TA79" Return
VCS EDIT>
```

This example shows the tape drive 2MUA0 being identified as a TA79 device.

MODIFY SCAN

Edits the information of an existing entry in the VCS configuration describing a scan profile.

Format

MODIFY SCAN *scan-name (/qualifier . . .)*

Parameters

scan-name
is the name of the scan profile.

Qualifiers

/EVENT=(event-name(, . . .))
/NOEVENT=(event-name(, . . .))
is a list of events to be added to or removed from the list of events making up the scan profile. The new events are appended to the list of events. Use the ADD EVENT command to create the entry for an event in the configuration.

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the scan profile.

/SCAN=(scan-name(, . . .))
/NOSCAN=(scan-name(, . . .))
is a list of other scan profiles to be added to or removed from the list of events making up the scan profile. The new scan profiles are appended to the list of scan profiles. Recursive scan profile inclusion is not allowed.

MODIFY SCAN

Description

The MODIFY SCAN command allows you to edit the information of an existing entry describing a scan profile. See the ADD SCAN command for more information.

Example

```
VCS EDIT> MODIFY SCAN ALPHA /NOEVENT=CAD_SOFTWARE_MESSAGES Return
VCS EDIT>
```

This example shows the removal of event CAD_SOFTWARE_ MESSAGES from the scan profile ALPHA.

MODIFY USER

Edits the information of an existing entry in the VCS configuration describing an authorized user's access privileges.

Format

MODIFY USER *user-name (/qualifier . . .)*

Parameters

user-name
is the name of an authorized user.

Qualifiers

/AUTHORIZED_ACCESS=({ ***node-name(, . . .)*** / ***cluster-name(, . . .)*** } ***)***

/NOAUTHORIZED_ACCESS=({ ***node-name(, . . .)*** / ***cluster-name(, . . .)*** } ***)***

is a list containing the names of the individual serviced nodes, or those represented by a defined VAXcluster name, to be added to or removed from the currently defined list of serviced nodes which the user is authorized to access. Use the ADD NODE command to create the entry for a serviced node in the configuration. Use the ADD CLUSTER command to create the list of node members of a VAXcluster.

You may use the wildcard character, asterisk (*), to indicate all serviced nodes to be added or removed.

/INFORMATION=site-specific-information
is a quoted text string of up to 255 characters in length containing site-specific information. This field could be used to contain a description of the user's operations duties and organization affiliates.

/MANAGEMENT
/NOMANAGEMENT
indicates if the user is authorized to perform VCS management functions.

MODIFY USER

Description

The MODIFY USER command allows you to edit the information of an existing entry describing the access restrictions imposed upon a user. See the ADD USER command for more information.

Example

```
VCS EDIT> MODIFY USER VOBA - Return
_VCS EDIT> /INFORMATION="Digital Customer Service Engineer" - Return
_VCS EDIT> /AUTHORIZED=HSC001 Return
VCS EDIT>
```

This example shows the user VOBA, who is a Digital Customer Service engineer, being authorized to access the HSC node HSC001.

QUIT

Terminates an editing session without saving configuration changes.

Format

QUIT

Description

The QUIT command terminates an editing session immediately without saving the configuration changes. No verification of the configuration data consistency is performed either.

Example

```
VCS EDIT>  QUIT  Return
$
```

The QUIT command immediately terminates the editing session and returns the user to DCL.

SET

Defines or changes characteristics of the VCS operating environment.

Format

SET *option (/qualifier . . .)*

Parameters

option
is a keyword denoting an option in the VCS configuration. The valid options are:

- DISPLAY
- LOGGING
- SECURITY
- VERIFY

Description

Use the SET command to change the characteristics of a VCS option.

SET DISPLAY

Defines the DECwindows display locations of the VCS Central Control Coordinator interfaces.

Format

SET DISPLAY */qualifier*

Qualifiers

/SERVERS=(display(, . . .))
/NOSERVERS=(display(, . . .))
is a list of DECwindows displays. Each display is specified in the format of *node::server.screen*.

where:

node::	is the DECwindows workstation, accessible over a DECwindows transport mechanism (such as DECnet), on which you want to display the C³ interface. *Node* cannot be a cluster alias (a name that represents multiple nodes configured in a VAXcluster), but must identify an individual node.
server.screen	specifies the DECwindows display server and display monitor.

You may specify a maximum of 8 different displays. The /SERVERS qualifier adds the specified display to the current list. The /NOSERVERS qualifier deletes the specified display from the current list.

The default DECwindows display is 0::0.0, which is the VCS host system.

SET DISPLAY

Description

The SET DISPLAY command allows you to define the DECwindows display locations for the Central Control Coordinator interfaces. At system startup, on the VCS host system, as many Central Control Coordinator interfaces as there are display servers will be activated. Each will have a DECwindows workstation assigned as its display destination.

Use the SHOW CHARACTERISTICS command to display the current information about the display locations.

Example

```
VCS EDIT> SET DISPLAY /SERVERS=("BETA::0.0","RHO::0.1") Return
VCS EDIT>
```

This example shows nodes BETA and RHO set to be the VCS display servers.

SET LOGGING

Defines the VCS logging and archiving characteristics.

Format

SET LOGGING */qualifier . . .*

Qualifiers

/ARCHIVE=file-name
is the file name of a DCL procedure. If you do not specify a file type, the Editor uses the default file type of COM. No wildcard characters are allowed in the file specification.

This DCL procedure is to be executed when the disk is full at the level specified by the /WARN qualifier. If no procedure is specified, when the /DELETE level is reached, VCS deletes the oldest log files in the current log directory to accommodate the newly received data.

The default archiving procedure is SYS$MANAGER:VCS$ARCHIVE.COM.

/DELETE=percentage
is a percentage of the disk device where the VCS log directory is located. When unused space on the disk device falls below this percentage, VCS deletes the oldest console and event log files.

The default percentage is 1%.

/DIRECTORY=(directory-name(, . . .))
is a list of disk directories to form the VCS log directory search list. The new log files are created in the first directory of the search list. The older log files can reside in any directory of the search list.

The default log directory is SYS$MANAGER.

/WARN=percentage
is a percentage of the disk device where the VCS log directory is located. When unused space on the disk device falls below this percentage, VCS issues a warning. If a DCL procedure is specified by the /ARCHIVE qualifier, it is executed.

The default percentage is 2%.

Description

The SET LOGGING command allows you to define the VCS logging and archiving characteristics. Use the SHOW CHARACTERISTICS command to display the current logging and archiving information.

If verification is enabled, the following items are validated:

- The specified log directories must exist on the VCS host system.
- The disk full warning percentage must be greater than the disk full delete percentage.

If you specify SYS$MANAGER:VCS$ARCHIVE.COM, supplied by Digital, to be the log file archiving command procedure, the following actions are performed when the disk is full at the level specified by the /WARN qualifier.

1 VCS sends a mail message, similar to the following one, to the SYSTEM account:

```
From: SYSTEM
To: SYSTEM
Subj: %VCS-W-FULLWARN, Disk full warning. Files will be DELETED if action is
not taken!

The VCS warning threshold has been reached for disk usage, please clean up
the log files in the log file directory.

Log file directory:
Device:                        _$2$DUA2:
Total blocks:                      891072
Number of free blocks:             452133
In use:                            438939

If space is not freed and the delete threshold is reached, VCS will delete
the oldest log files until an acceptable amount of space is availaible
on the log file directory device.

Log files based on logical name VCS$LOG
contained in:

Directory SYS$SYSROOT:[SYSMGR]
```

```
VCS$1989_08_01.LOG;1                    125/125

Total of 1 file, 125/125 blocks.


Event log files based on logical name VCS$LOG
contained in:

Directory SYS$SYSROOT:[SYSMGR]

VCS$EVENT_1989_08_01.LOG;1              67/67

Total of 1 file, 67/67 blocks.

 -- VAXcluster Console System (VCS)
```

2 A message, similar to the following one, is broadcasted to all active terminals logged into the VCS host system repeatedly at intervals of 30 seconds:

```
*URGENT* message on MISVCS from user SYSTEM at _MISVCS$TWA9:   15:34:34
%VCS-W-FULLWARN, Disk full warning!  Please free space on logging disk.
```

Example

```
VCS EDIT>  SET LOGGING /DIRECTORY=(DUA0:[VCS],DFSC1001:[ARCHIVE]) -Return
_VCS EDIT>  /ARCHIVE=SYS$MANAGER:VCS_ARCHIVE.COM Return
VCS EDIT>
```

This example shows the VCS log directory search list containing a local disk directory as the logging directory for the newly created log files, and a DFS disk directory as a secondary storage location for older log files. The archiving DCL command procedure SYS$MANAGER:VCS_ARCHIVE.COM is to be activated (moving files from DUA0:[VCS] to DFSC1001:[ARCHIVE]) when the logging disk DUA0 has 1% of free space left.

SET SECURITY

Activates the VCS security facility.

Format

SET (NO)SECURITY

Description

The VCS security facility is defined to be the control of user access to VCS and its serviced entities. You use the ADD USER command to describe this access control. The SET SECURITY command allows you to activate these access restrictions. The SET NOSECURITY deactivates the security facility but leaves the access control definitions intact. Use the SHOW CHARACTERISTICS command to display the current security status.

The default is SET NOSECURITY.

Example

```
VCS EDIT> SET SECURITY Return
VCS EDIT>
```

This command enables the security facility.

SET VERIFY

Activates the VCS data validation facility.

Format

SET (NO)VERIFY

Description

The SET NOVERIFY command allows you to temporarily deactivate the data verification facility in the Configuration Editor. This allows you to run the Editor and create an otherwise legal VCS configuration on a VMS system that is not intended to be the VCS host system. Use the SHOW CHARACTERISTICS command to display the current verification status.

Note that VCS does not reconfigure or restart with an unverified configuration file.

The Editor always starts with SET VERIFY.

Example

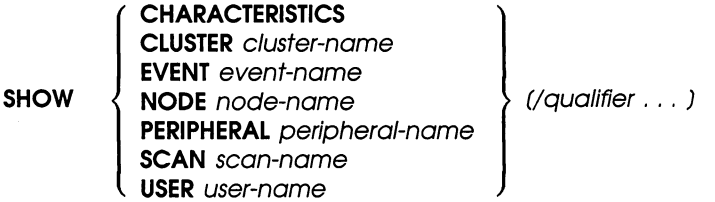
```
VCS EDIT>  SET NOVERIFY [Return]
VCS EDIT>
```

This command temporarily disables the verification facility.

SHOW

Displays information about a specified entity or the current characteristics of the VCS operating environment.

Format



Parameters

CHARACTERISTICS
displays the information about the configuration file itself and the parameters defined by the SET commands.

cluster-name
is the name of the VAXcluster. You may use the wildcard character, asterisk (*), to indicate all existing VAXclusters in the configuration.

event-name
is the name of the event. You may use the wildcard character, asterisk (*), to indicate all existing events in the configuration.

node-name
is the name that you assign to a serviced node in the VCS configuration. You may use the wildcard character, asterisk (*), to indicate all existing nodes in the configuration.

peripheral-name
is the name of the peripheral. You may use the wildcard character, asterisk (*), to indicate all existing peripheral devices in the configuration.

scan-name
is the name of the scan profile. You may use the wildcard character, asterisk (*), to indicate all existing scan profiles in the configuration.

user-name
is the name of an authorized user. You may use the wildcard character, asterisk (*), to indicate all existing authorized users in the configuration.

Qualifiers

/APPEND
indicates that the displayed information is to be appended to the file specified by the /OUTPUT qualifier.

/BRIEF (D)
indicates if an abbreviated description of the specified option is to be displayed.

This is the default.

/CREATE (D)
indicates that a new copy of the file, as specified by the /OUTPUT qualifier, is to be created to contain the displayed information.

This is the default.

/FULL
indicates if a full description of the specified option is to be displayed.

The default is /BRIEF.

/OUTPUT=file-name
Specifies the output file for the displayed information. If you do not enter the qualifier, or if you enter /OUTPUT without a file name, the output is sent to SYS$OUTPUT.

SHOW

Description

The SHOW command allows you to interrogate the Editor for information about any current definition of the various components in the VCS configuration. The /FULL and /BRIEF qualifiers are meaningful only when they are used to display the information about the entities (such as nodes, scan profiles, and events) created in the configuration by the ADD command.

Example

```
VCS EDIT>  SHOW NODE ALPHA /FULL /OUTPUT=ALPHA.LIS Return
```

This example shows the display of information about node ALPHA being directed to the output file ALPHA.LIS.

SPAWN

Spawns a subprocess suspending, but not ending, your current editing session.

Format

SPAWN *(DCL-command)*

Parameters

DCL-command
is a DCL command that you want to execute by the subprocess.

Description

The SPAWN command creates a subprocess of your current editing session. This is useful for running another utility, such as LATCP, without having to end your editing session.

Examples

1
```
VCS EDIT>  SPAWN Return
$
```

This example shows the subprocess is spawned from the current editing session and stays in DCL waiting for the user's command.

2
```
VCS EDIT>  SPAWN DIRECTORY Return

Directory USER3:[VOBA.VCS.SPECIFICATION]

FS_EDITOR.SDML;103
FS_FRONT.SDML;45
FUNCTIONAL.PS;3
FUNCTIONAL.SDML;3
FUNCTIONAL.TXT;14
FUNCTIONAL_CONTENTS.PS;2
FUNCTIONAL_CONTENTS.TXT;12
PROJECT.SDML;1

Total of 8 files.
VCS EDIT>
```

This example shows the DCL command DIRECTORY being executed in a subprocess spawned from the current editing session. The editing session is then resumed after the DCL command execution.

VERIFY

Validates the current VCS configuration information for consistency.

Format

VERIFY

Description

Use the VERIFY command to validate the VCS configuration data that you have entered thus far.

Example

```
VCS EDIT> VERIFY Return
VCS EDIT>
```

This command verifies the current VCS configuration information for consistency and let you proceed with the editing session.

5

DCL Commands

This chapter describes the VCS DCL commands. These commands are used to activate both the processes (for example, the I/O Data Logger) at system startup, and the user interfaces and utilities (for example, the Connect Interface, Configuration Editor, and so on) interactively. In addition, a number of VCS functions are also made available at DCL level.

You can abbreviate commands, as long as you enter enough characters to make them unique to VCS. A VCS command may also require a list of one or more node names or a component.

5.1 Node Name Conventions

Some VCS commands can operate on one or more nodes; other VCS commands can operate only on a single node. All VCS commands that require node names also accept logical names. The rules for specifying node names are as follows:

- The node names recognized by VCS are those names listed in the VCS configuration file. The REVIEW command, however, allows you to review data for nodes which may no longer be in the configuration.
- Any VCS command that accepts a single node name also accepts a logical name that equates to a node name. For example, you can do the following:

```
$ DEFINE NODE1 PEPPER
$ VCS CONNECT NODE1
```

or

```
$ VCS CONNECT PEPPER
```

- Any VCS command that accepts a list of node names also accepts a list of logical names, each of which equates to a node name or a single logical name that equates to a list of node names or any combination thereof. For example, you can define the following logical names:

```
$ DEFINE NODE1 PEPPER
$ DEFINE NODE2 HEART
$ DEFINE NODE3 HSC001
$ DEFINE NODE4 HSC002
$ DEFINE ALL_VAXEN NODE1,NODE2
$ DEFINE ALL_HSCS NODE3,NODE4
$ DEFINE ALL_NODES ALL_VAXEN,ALL_HSCS
```

  Any of the following VCS commands allow you to monitor the same set of nodes:

```
$ VCS MONITOR PEPPER,HEART,HSC001,HSC002
$ VCS MONITOR NODE1,NODE2,NODE3,NODE4
$ VCS MONITOR ALL_VAXEN,ALL_HSCS
$ VCS MONITOR ALL_VAXEN,HSC001,NODE4
$ VCS MONITOR ALL_NODES
```

- You can abbreviate node names (except for the REVIEW command). However, you cannot abbreviate logical names.
- The default node name for VCS ACCESS, VCS COORDINATE, VCS MONITOR, VCS RECORD, and VCS REVIEW is VCS$ALL_NODES, which is a logical name that specifies all the nodes in the current VCS configuration.

5.2 DCL Command Reference

The VCS DCL commands described in this section are listed in Table 5–1.

Table 5–1 VCS DCL Commands

ACCESS	EDIT	REVIEW
CONNECT	MONITOR	SHUTDOWN
CONVERT	RECONFIGURE	START
COORDINATE	RECORD	

VCS

Starts a VCS process or user interface, or executes a VCS function.

Description

The VCS command starts all VCS processes and user interfaces. Certain VCS processes, such as the I/O Data Logger, Event Logger and Data Scanner, are started during system startup. You use the VCS command to start the user interfaces interactively. A number of VCS functions such as reconfiguration or shutdown can also be invoked from DCL with the VCS command.

VCS ACCESS

Activates the Access Interface to execute a user-written application or a default application provided by VCS.

Format

VCS ACCESS *node-name(, . . .) (/qualifier . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

The default is a list of node names defined by the logical name VCS$ALL_NODES, which includes all the nodes in the current VCS configuration.

Qualifiers

/APPLICATION=file-name
is the name of your shareable image file. You use the /APPLICATION qualifier to indicate to the Access Interface that the specified user-written application is to be executed. The file name is either a logical name that translates to a complete file specification, or the name of a file to which "SYS$SHARE:.EXE" is applied. In either case, the file must be an installed shareable image with the same privileges as the executable image VCS$IFEX.EXE. If you define a logical name for the application, this logical name must be a system-wide, executive-mode logical name.

/APPLICATION and /DEFAULT are mutually exclusive.

/DEFAULT (D)
indicates the Access Interface Default application, contained in the shareable image SYS$SHARE:VCS$ACCESS_SECTION.EXE, is to be executed. /DEFAULT and /APPLICATION are mutually exclusive.

This is the default.

/DISPLAY=display-name
indicates the DECwindows display server name where the application interface is going to reside. The display name is specified in the format of *node::server.screen*.

where:

node::	is the DECwindows workstation, accessible over a DECwindows transport mechanism (for example, DECnet), on which you want to display the application interface. *Node* cannot be a cluster alias (a name that represents multiple nodes configured in a VAXcluster), but must instead identify an individual node.
server.screen	specifies the DECwindows display server and display monitor.

This qualifier is applicable only when /INTERFACE=DECWINDOWS is specified.

The default DECwindows display is 0::0.0, which is the VCS host system.

/INTERFACE= { ***CHARACTER_CELL*** / ***DECWINDOWS*** }
indicates whether the application DECwindows or the character-cell terminal interface is to be activated. If this qualifier is omitted, the following steps are used to determine whether the DECwindows or character-cell terminal interface is to be activated.

1. If the device class of SYS$OUTPUT is DC$_WORKSTATION, the DECwindows interface is activated.
2. If the logical name DECW$DISPLAY is defined, the DECwindows interface is activated.
3. Activate the character-cell terminal interface.

VCS ACCESS

Description

The VCS ACCESS command activates the Access Interface, which enables you to execute your own software to enhance the processing of console events. To assist you in your application development, VCS also provides the means to debug your code using the VAX/VMS Debugger.

To activate your user-written application, you must specify the file name that points to the shareable image of your software with the /APPLICATION parameter. The *callback routines* you need to provide in your shareable image when interfacing with the Access Interface are described in Chapter 9. If you specify a user-written application, when an event is detected, the Access Interface passes the program flow control to your callback routines (along with information about that event) for additional processing.

VCS also provides a default application you can use with the Access Interface. If you specify the /DEFAULT qualifier, the Access Interface Default Application is activated instead. It receives the VCS events and scan events and displays them at the specified output device.

Example

```
$ DEFINE/SYSTEM/EXECUTIVE VCS$DISPLAY SYS$SHARE:VCS$DISPLAY.EXE Return
$ VCS ACCESS /APPLICATION=VCS$DISPLAY Return
```

These commands activate the user-written application VCS$DISPLAY.EXE.

VCS ACCESS /DEFAULT

The Access Interface Default Application receives the VCS events and scan events, and displays them at the specified output device.

Format

VCS ACCESS /DEFAULT *node-name(, . . .) (/qualifier . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

The default is a list of node names defined by the logical name VCS$ALL_NODES, which includes all the nodes in the current VCS configuration.

Qualifiers

/HEADER (D)
/NOHEADER
indicates whether the column headings will be displayed prior to the events.

The default is /HEADER.

/OUTPUT=file-name
indicates where the events and their associated information are to be displayed. If you do not enter the qualifier, or if you enter /OUTPUT without a file name, the output is sent to the current process default output stream or device, identified by the logical name SYS$OUTPUT.

VCS ACCESS /DEFAULT

/TIME_STAMP (D)
/NOTIME_STAMP
indicates if the time stamp associated with each event is to be included in the displayed information at the output destination specified by the /OUTPUT qualifier.

The default is /TIME_STAMP.

Description

The Access Interface Default Application receives the VCS events and scan events, and displays them at the specified output device. For each event, the following information is displayed:

- The time at which it occurred.
- The system from which it was generated. **VCS** if it was a VCS event.
- The priority at which it was defined by the system manager in the scan file. An asterisk * is displayed if it was a VCS event.
- The event name. **EVENT** if it was a VCS event.
- The line of console data containing the scan event text. Non-printable characters in the line of text are displayed as dot characters. An explanatory message is displayed if it was a VCS event.
- For scan events, a second line of data appears under the "Information" column if the scan file's event record contains any user information associated with the event.

The following is a sample display of the Access Interface Default application.

```
    Date    /  Time    System Pri Event  Information
_____________ ________ ______ ___ _____ ____________________________________
15-Oct-1987 14:40:44 NODE3   10 Login  Username: VOBA
                                       Attempted login at console terminal
15-Oct-1987 14:42:23 VCS      * EVENT  System NODE3 has been locked by user HOFFMAN
```

The first line in the display represents a scan event message. It is triggered by a match on the string **Username:** in one of the scan files associated with NODE3. In the scan file, the event has been assigned a priority of 10 and the event name of Login.

The third line in the display represents a VCS event generated by a user action. In this case, the user HOFFMAN has entered the Monitor Interface command SET LOCK ON to deny other users access to NODE3. For this type of event, VCS provides a system name of VCS, a priority of *, and an event name of EVENT.

Example

```
$ VCS ACCESS /DEFAULT HSC70 /OUTPUT=SYS$MANAGER:HSC70.EVENTS Return
```

This command activates the Access Interface Default Application to write only those events which occur on node HSC70 out to the ASCII text file, SYS$MANAGER:HSC70.EVENTS. This command is used for tracking a problem that is occurring on only one node.

VCS CONNECT

Activates the Connect Interface to connect your terminal to the node you specify and allows you to communicate with that system as if you were on a local console terminal.

Format

VCS CONNECT *node-name (/qualifier . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names.

Qualifiers

/BREAK_KEY=key-name
/NOBREAK_KEY (D)
is a control key on your keyboard that is to be used when you want to send a Break signal to the node to which you are connected. This key is not the same as the Break key. The key name is the name of a control key in the form CTRL_*x*_KEY, where *x* is a letter from A to Z; or BS_KEY; or LF_KEY.

/NOBREAK_KEY indicates that there is no way to send a Break signal to the connected node.

The default is /NOBREAK_KEY.

/CONTINUE
/NOCONTINUE
indicates whether the interface continues to accept input from SYS$COMMAND after the contents of the /INPUT files are successfully transmitted. This qualifier is ignored if you did not specify the /INPUT qualifier.

Use /NOCONTINUE to issue an implicit escape key immediately after the contents of the last file is transmitted successfully.

The default for an interactive session is /CONTINUE. The default for batch mode is /NOCONTINUE.

/ESCAPE_KEY=key-name
is a control key on your keyboard that is to be used when you want to terminate the connection to the node. This key is not the same as the Esc key. The key name is the name of a control key in the form CTRL_*x*_KEY, where *x* is a letter from A to Z; or BS_KEY; or LF_KEY.

The default escape key is Ctrl/G.

/INPUT=file-name(, . . .)
is an ASCII text file whose content is to be transmitted. When you specify a list of files, separate the names with commas. If any of the files in the list do not exist, none of them will be transmitted.

If a file contains a control character that is identical to the one you specified with the /ESCAPE_KEY qualifier, VCS sends the character to the node as a regular control character. Typing the escape character at the controlling terminal while the files are being sent will halt the transmission temporarily. VCS prompts you with:

```
Do you want to exit (Yes/No)?
```

If you answer no, the transmission continues.

If a file contains a control character that is identical to the one you specified with the /BREAK_KEY qualifier, VCS translates the character into a Break signal on the console line.

/LOCK
/NOLOCK (D)
indicates whether you intend to reserve the exclusive input rights to the connected node's console line. If you do so, other users cannot send data to the node, but they can still examine log information and view your current activity. The node remains locked until you disconnect from the node by pressing the escape character. /NOLOCK means that you share input rights with other users on the specified console line. Any other operator is able to enter data simultaneously on the console line.

The default is /NOLOCK.

/RATE= { ***FAST*** | ***MEDIUM*** | ***SLOW (D)*** }
is the rate at which characters are sent from the /INPUT file to the nodes. FAST indicates that the contents of the files will be transmitted as fast as possible to the node. MEDIUM indicates that the contents of the files will be transmitted to the node at about 8 to 10 characters per second. SLOW indicates that the contents of the files will be transmitted to the node at about 1 to 3 characters per second.

The /RATE qualifier does not change your baud rate. The /RATE qualifier is ignored if you do not specify the /INPUT qualifier.

The default rate is SLOW.

Description

The VCS CONNECT command connects your terminal to the node you specify. The node controls your full terminal screen. You may want to use this command to run a program that requires full-screen control, or when you want the full terminal screen for your work.

When you type the VCS CONNECT command, VCS displays a message in one of the two formats similar to the following messages:

```
%VCS-I-SHOWNBKESC, No break is set, CTRL/G is escape
%VCS-I-SHOWBRKESC, CTRL/B is break, CTRL/G is escape
```

The %VCS-I-SHOWNBKESC message means that you have not defined a break character and the default escape character is Ctrl/G. If you use the /ESCAPE_KEY qualifier with the VCS CONNECT command to define your own escape character, VCS displays the character you defined. If you use the /BREAK_KEY qualifier with the VCS CONNECT command to define a break character, VCS displays the %VCS-I-SHOWBRKESC message which includes both the break and escape characters.

Once you have a connection to the specified node through the console line, characters (excluding the substitute break and escape characters, Ctrl/S and Ctrl/Q), are transmitted directly to the node. Output from the node is sent to your terminal untranslated. This continues until you type one of the designated control characters and you are prompted to verify the action with one of the following:

```
Do you want to exit (Yes/No)?
Do you want to transmit a break (Yes/No)?
```

If you answer no, the actual control character entered is transmitted to the console. If you answer yes, the requested action occurs.

If you define both the substitute break and escape characters to be the same, the escape action takes precedence over the break action. For example, if you define both characters to be CTRL_G_KEY, when you press Ctrl/G, VCS asks whether you intend to exit the Connect Interface. However, during the session, you will not be able to transmit any Break signal using Ctrl/G.

Note *Any text you type on a terminal connected to a console line is echoed on all terminals connected to the same console line with the MONITOR or CONNECT commands.*

If the VCS CONNECT/INPUT command is executed in a batch command procedure, the following message will be displayed:

```
%VCS-I-HALFDUPLEX, Half-duplex connection
```

The interface will not display any console data received from the connected node. They are still being displayed on all terminals connected to the same console line with the MONITOR or CONNECT commands.

Example

```
$ VCS CONNECT/LOCK MINNIE Return
%VCS-I-SHOWNBKESC, No break is set, CTRL/G is escape
```

This example connects to the node named MINNIE from DCL command level. When you connect to the node, VCS displays a message that shows your current break character and escape character. Now, you can enter commands to MINNIE.

After you finish your work on MINNIE, press the escape character to exit to DCL level. VCS prompts with a question to make sure you want to disconnect. Press Y to disconnect or N if you do not want to disconnect from MINNIE.

VCS CONVERT

Converts data into forms and formats appropriate for processing by the VAXcluster Console Software components.

Format

VCS CONVERT { **/DATA** *file-name(, . . .) (/qualifier)* } { **/ICON** *(file-name(, . . .)) (/qualifier)* }

Parameters

file-name
is the file that contains the data to be converted.

Qualifiers

/APPEND
indicates that the converted data is to be appended to the file specified by the /OUTPUT qualifier.

/CREATE (D)
indicates a new copy of the file, as specified by the /OUTPUT qualifier, is to be created to contain the converted data.

This is the default.

/DATA
indicates configuration and scan data conversion is to be performed. The /DATA and /ICON qualifiers are mutually exclusive.

/ICON (D)
indicates DECwindows Digital Document Interchange Format (DDIF) icon bitmap conversion is to be performed. The /ICON and /DATA qualifiers are mutually exclusive.

This is the default.

/INVERT
/NOINVERT (D)
indicates that the foreground and background bitmap representation is to be inverted. When an image is captured on the screen, it is assumed that the lighter screen color is the background and the darker color is the foreground. If /INVERT is specified, the lighter color is treated as the foreground and the darker color as the background. The /INVERT qualifier is only applicable when used with the /ICON qualifier.

The default is /NOINVERT.

/LOG
/NOLOG (D)
indicates progress is to be reported during the conversion process.

The default is /NOLOG.

/OUTPUT=file-name
is the file to contain the converted data. No wildcard characters are allowed in the file specification.

The default file name for icon conversion is VCS$DW_ICON_DFN.UIL. The default file name for data conversion is SYS$MANAGER:VCS$CONFIGURATION.DAT.

/REVIEW
/NOREVIEW (D)
indicates that the icon conversion result is to be displayed on the VAXstation monitor. The /REVIEW qualifier is only applicable when used with the /ICON qualifier.

The default is /NOREVIEW.

/STATISTICS
/NOSTATISTICS (D)
indicates statistics are to be reported at the end of the conversion process.

The default is /NOSTATISTICS.

/UID_FILE=file-name
is the UID file containing the previously converted icons to be reviewed. The /UID_FILE is only applicable when used with the /REVIEW qualifier. You may not specify both the /UID_FILE qualifier and an input file for conversion and review.

The default file name is VCS$DW_ICON_DFN.UID.

Description

The VCS CONVERT /ICON command allows you to create customized icons to be used by the VCS DECwindows interfaces. See Chapter 8 for procedural information.

The VCS CONVERT /DATA command is invoked by the VCS software installation procedure to convert VCS Version 1.2 data into a form usable by VCS Version 1.3. This command creates a new configuration file in binary format for all information that used to be in separate VCS Version 1.2 configuration and scan files, and the rights list data base.

The VCS CONVERT /DATA command also creates an indirect command file, with the same name as the output file but having the .COM file extension in the default directory. This indirect command file contains the appropriate VCS Configuration Editor commands to reproduce the new configuration file from scratch. The VCS Configuration Editor requires that the scan events must have been defined in the configuration file before they are referenced in association with a node. Thus, the order of the converted files on the command line is significant because that dictates the order of the commands being created in the indirect command file. By specifying the scan files first for conversion, the new equivalent VCS Configuration Editor commands are generated first in the indirect command file.

Examples

1
```
$ VCS CONVERT/ICON/REVIEW VCS$ICONS:*.DDIF,NEWCPU.DDIF Return
```

This example converts the newly created DDIF icon bitmap file NEWCPU.DDIF in the default directory. It also converts the currently available set of default icons from the VCS$ICONS directory. All the icons are then displayed for review on the VAXstation monitor display.

2
```
$ COPY SYS$MANAGER:VCS$CONFIGURATION.DAT SYS$MANAGER:*.DAT_012 Return
$ VCS CONVERT/DATA/OUTPUT=SYS$MANAGER:VCS$CONFIGURATION.DAT - Return
_$ SYS$MANAGER:VCS$SCAN*.DAT,VCS$CONFIGURATION.DAT Return
```

This example shows the commands used by the software installation procedure to create a backup copy of the current configuration file having the same file name but with extension .DAT_012 before the actual conversion. The new configuration file is put back into SYS$MANAGER after the conversion. The command also creates an indirect command file called VCS$CONFIGURATION.COM in the default directory, which contains all of the VCS Configuration Editor commands that could be used to reproduce the converted configuration file from scratch.

VCS COORDINATE

Activates the Central Control Coordinator Interface to monitor, manage, and interact with the nodes you list on the command line.

Format

VCS COORDINATE *node-name(, . . .) (/qualifier . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names.

Qualifiers

/ACCESS_MODES= { ***MONITOR*** / ***INTERACTIVE (D)*** }
indicates whether interactive console interfaces are allowed to be activated from this C³ session. The /ACCESS_MODES=MONITOR qualifier disables the C³ Connect and Set Host commands. The /ACCESS_MODES=MONITOR qualifier also prevents the Monitor Interface from being activated in INTERACTIVE mode from the C³.

The default is /ACCESS_MODES=INTERACTIVE.

/DISPLAY=display-name
indicates the DECwindows display server name where you want the Central Control Coordinator Interface to reside. The display name is specified in the format of *node::server.screen*.

where:

node::	is the DECwindows workstation, accessible over a DECwindows transport mechanism (for example, DECnet), on which you want to display the C^3 interface. *Node* cannot be a cluster alias (a name that represents multiple nodes configured in a VAXcluster), but must instead identify an individual node.
server.screen	specifies the DECwindows display server and display monitor.

The default DECwindows display is 0::0.0, which is the VCS host system.

Description

The VCS COORDINATE command starts the Central Control Coordinator Interface on a specific DECwindows VAXstation to allow you to monitor, manage and interact with the specified nodes.

Examples

1
```
$ SET DISPLAY /CREATE /NODE=CHI Return
$ VCS COORDINATE Return
```

These commands start a Central Control Coordinator Interface and direct VCS to display the C^3 windows at the VAXstation node CHI.

2
```
$ VCS COORDINATE /DISPLAY=CHI::0.0 /ACCESS_MODES=MONITOR Return
```

This command starts a Central Control Coordinator Interface displayed at node CHI. The CONNECT and SET HOST commands are not available at node CHI.

3 `$ VCS COORDINATE ALPHA,BETA` Return

This command starts a Central Control Coordinator Interface which can only interact with nodes ALPHA and BETA.

VCS EDIT CONFIGURATION

Starts the Configuration Editor.

Format

VCS EDIT CONFIGURATION *(/qualifier . . .)*

Qualifiers

/DISPLAY=display-name
indicates the DECwindows display server name where the Editor DECwindows interface is going to reside. The display name is specified in the format of *node::server.screen*.

where:

node::	is the DECwindows workstation, accessible over a DECwindows transport mechanism (for example, DECnet), on which you want to display the Editor. *Node* cannot be a cluster alias (a name that represents multiple nodes configured in a VAXcluster), but must instead identify an individual node.
server.screen	specifies the DECwindows display server and display monitor.

This qualifier is applicable only when /INTERFACE=DECWINDOWS is specified.

The default DECwindows display is 0::0.0, which is the VCS host system.

/INPUT=file-name
is the VCS configuration file whose contents are to be updated. No wildcard characters are allowed in the file specification. If the input file does not exist, a new one will be created when the Configuration Editor is terminated with the EXIT command.

The default file name is SYS$MANAGER:VCS$CONFIGURATION.DAT.

/INTERFACE= { ***CHARACTER_CELL*** | ***DECWINDOWS*** }

indicates whether the DECwindows or the character-cell terminal interface is to be activated. If this qualifier is omitted, the following steps are used to determine whether the DECwindows or character-cell terminal interface is to be activated.

1 If the device class of SYS$OUTPUT is DC$_WORKSTATION, the DECwindows interface is activated.

2 If the logical name DECW$DISPLAY is defined, the DECwindows interface is activated.

3 Otherwise, the character-cell terminal interface is activated.

/OUTPUT=file-name

is the new VCS configuration file to be created. No wildcard characters are allowed in the file specification.

The default file name is the same as that specified by the /INPUT qualifier.

Description

The Configuration Editor allows you to create and edit the VCS configuration information. The VCS configuration is composed of information describing its environment and the different entities it services. See Chapters 3 and 4 for descriptions about how to use the Configuration Editor.

Example

```
$ VCS EDIT CONFIGURATION Return
VCS EDIT>
```

This command starts the Configuration Editor.

VCS MONITOR

Starts a Monitor Interface session in which you can access the nodes you list on the command line. The Monitor Interface display shows current log information for the first node listed on the command line.

Format

VCS MONITOR *node-name(, . . .) (/qualifier . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

The default is a list of node names defined by the logical name VCS$ALL_NODES, which includes all the nodes in the current VCS configuration.

Qualifiers

/COMMAND=file-name
/NOCOMMAND (D)
is a command file that contains the Monitor Interface commands. Use a text editor to create an ASCII file that contains the Monitor Interface commands. Then use the /COMMAND qualifier to execute the commands in the file at the start of a session.

/NOCOMMAND indicates that no command file is to be processed.

The default is /NOCOMMAND.

/ACCESS_MODES= { ***MONITOR*** / ***INTERACTIVE (D)*** }
indicates whether the CONNECT, SELECT, and OUTPUT commands are to be disabled during the Monitor Interface session. The qualifier /ACCESS_MODES=MONITOR may be used when the Monitor Interface is activated from a captive account to allow the user to view console data only.

The default is /ACCESS_MODES=INTERACTIVE.

/EVENT (D)
/NOEVENT
indicates whether the event summary window is displayed on the screen as part of the Monitor Interface display. If you specify /NOEVENT, the event indicators are not displayed on the screen when an event occurs and you cannot execute the Monitor Interface VIEW/EVENT command. The log window takes the two lines that otherwise would be used by the event summary window. The default is /EVENT.

/SECTION=file-name
is the TPU section file that contains your custom key definitions to be used during the Monitor Interface session. You can use the DEFINE/KEY and SAVE commands to create key definitions and save them in a file. Then use the /SECTION qualifier to execute the file when you start a Monitor Interface session.

The default file name is SYS$LIBRARY:VCS$MONITOR_SECTION.TPU$SECTION.

VCS MONITOR

Description

When you start the Monitor Interface, its display fills your video terminal screen. The log information about the first node you specified on the command line appears in the log window at the top of the screen. VCS may show information about other nodes you typed on the command line in the event summary window at the bottom of the screen. You can use various Monitor Interface commands to establish a communications path to a node or nodes, or to view log information about the nodes while you continue to monitor events in the event summary window.

Use the EXIT command (or Ctrl/Z) to stop a session and return to DCL level.

You must specify the nodes you want to access in the current session. If you do not specify a particular node name in the command line, you cannot access that node during that session.

See Chapter 6 for the complete description of the Monitor Interface.

Example

```
$ VCS MONITOR /COMMAND=VCSDEFS.COM - Return
_$ /SECTION=DOC2:[GREEN.VCS]MYFILE.TPU$SECTION VAX1,VAX2 Return
```

This command starts the Monitor Interface and executes the command file, VCSDEFS.COM, and the section file, MYFILE.TPU$SECTION. The Monitor Interface displays current log information about the node VAX1, since it is the first node name that was specified on the command line. VCS displays event information as it occurs for VAX2 in the event summary window. You can communicate with or examine log information for either VAX1 or VAX2.

VCS RECONFIGURE

Notifies VCS of changes that the system manager made in the VCS configuration file, and requests VCS to attempt to repair the LAT host initiated connections that were interrupted.

Format

VCS RECONFIGURE

Description

When VCS is started up, it reads in the configuration file contents and retains that information for managing the console lines and the various user interfaces. That information remains static to the VCS processes. You may, at any time, update the configuration file, using the Configuration Editor, without affecting VCS's current knowledge about its environment and the entities it services.

To notify VCS of the changes that you made to the configuration file, you must issue the VCS RECONFIGURE command after successfully updating the information. Otherwise, that information remains ineffective until VCS is shut down and restarted.

When you issue the VCS RECONFIGURE command, if the configuration file has not been changed since the last time VCS processed its contents, nothing changes. The VCS RECONFIGURE command affects the VCS operations as follows:

- If you changed the scan profile definition in the configuration file, the new definition is in effect when the reconfiguration completes successfully.
- If you changed a node name, the old node is no longer in the VCS configuration. The logical name VCS$ALL_NODES is redefined to reflect the new list of node names.

- If you changed the terminal device associated with a node, that node is considered logically unchanged. (For example, data from the new terminal line is appended to the previously recorded data from the old terminal line.)
- A disabled system retains its disabled status.
- VCS notifies any running interfaces that a reconfiguration has occurred. If a node has been added to the configuration, an interface must be started again to gain access to the new node.
- If placement of the icons in the Central Control Coordinator Interface has been changed, the new placement is reflected in the currently active displays. New configuration information is available to the C³ Show Information windows.
- Changes to the ENS configuration file become active.

VCS RECONFIGURE does not affect the definition of the logical name VCS$LOG. The new definition will not take effect until you shut down and restart the VCS software.

The VCS RECONFIGURE command also requests VCS to attempt to repair any LAT host initiated connections that were interrupted.

Example

```
$ VCS RECONFIGURE Return
```

This command request VCS to update its current configuration information with the new information from the configuration file SYS$MANAGER:VCS$CONFIGURATION.DAT.

VCS RECORD

Activates the Record Interface to log the console data to a hardcopy terminal that is directly attached to the VCS host system.

Format

VCS RECORD *node-name(, . . .) (/qualifier)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

The default is a list of node names defined by the logical name VCS$ALL_NODES, which includes all the nodes in the current VCS configuration.

Qualifiers

/OUTPUT=device-name
is the terminal that you want to record log information.

The terminal is the one at which you issue the VCS RECORD command.

/TIME_STAMP (D)
/NOTIME_STAMP
indicates whether a time stamp is to accompany each line of output. The time stamp indicates when VCS began to receive the line of information from the node. /NOTIME_STAMP suppresses printing of the time stamps.

The default is /TIME_STAMP.

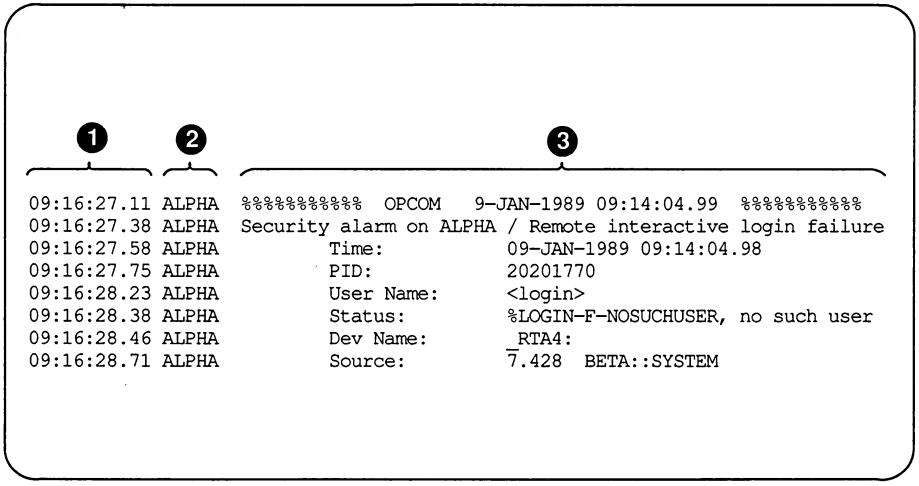
VCS RECORD

Description

Use the VCS RECORD command at a hardcopy terminal. The VCS RECORD command starts recording log information on the terminal from which you enter the command. The terminal continues to record log information until you press Ctrl/Y and enter the DCL EXIT command. If you enter the VCS RECORD command at a video terminal and direct output to a hardcopy terminal with the /OUTPUT qualifier, you cannot use the video terminal for other work until you stop the Record Interface. The Record Interface includes time stamps in the hardcopy log information by default. You control whether time stamps appear in the the hardcopy information with the /NOTIME_STAMP qualifier.

Figure 5–1 describes the console data as displayed by the Record Interface.

Figure 5–1 Recorded Text Anatomy


1
2
3
09:16:27.11 ALPHA %%%%%%%%%% OPCOM 9-JAN-1989 09:14:04.99 %%%%%%%%%%
09:16:27.38 ALPHA Security alarm on ALPHA / Remote interactive login failure
09:16:27.58 ALPHA Time: 09-JAN-1989 09:14:04.98
09:16:27.75 ALPHA PID: 20201770
09:16:28.23 ALPHA User Name: <login>
09:16:28.38 ALPHA Status: %LOGIN-F-NOSUCHUSER, no such user
09:16:28.46 ALPHA Dev Name: _RTA4:
09:16:28.71 ALPHA Source: 7.428 BETA::SYSTEM

MR-4078-RA

❶ The time stamp which indicates the time that VCS receives the first character of a line of information from the serviced node. Note, if the serviced node is a VAX/VMS system, the first character is usually the DCL prompt, which may be displayed hours before the first data character entered by the user at the console.

❷ The name of the serviced node which displayed the console text line.

❸ The text data from the serviced node.

Example

```
$ VCS RECORD PEPPER, HEART Return
```

This example shows how to record log information with time stamps for nodes PEPPER and HEART on the terminal where the command was issued. The following display is representative of a hardcopy log created with VCS RECORD.

```
10:35:26:01 PEPPER %%%%%%% OPCOM 30-MAR-1986 10:35:24.07 %%%%%%%
10:35:26:03 PEPPER Message from user DECNET on HEART
10:35:26:10 PEPPER DECnet starting
```

VCS REVIEW

Activates the Review Interface to allow you to examine the console and event data contained in the VCS console and event log files, for a specific set of serviced nodes over a specific period of time. The data you request can be output to a disk file or a terminal.

Format

VCS REVIEW *node-name(, . . .) (/qualifier)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. When you specify a list of node names or logical names, separate the names with commas.

There are two exceptions when you specify the node names for this command:

1 You cannot abbreviate the node names or the logical names.

2 You can specify **VCS** as a node name to see VCS data or events.

The default is a list of node names defined by the logical name VCS$ALL_NODES, which includes all the nodes in the current VCS configuration.

Qualifiers

/BEFORE=date-time
is the ending point in the time period for which data is to be retrieved. The time and date must follow the VMS convention and in the format:

```
[dd-mmm-yyyy[:]][hh:mm:ss.cc]
```

This time must not predate the available log files and must not be in the future. It must always be a later time than the time specified by the /SINCE qualifier. If /BEFORE is not specified, then the ending point of the time period defaults to the current date and time. If you specify a time without the date, it defaults to today at the specified time. If you specify a date without the time, it defaults to the specified date at 00:00:00.00.

/EVENT= [{ ***CONTEXT*** }]
[{ ***NOCONTEXT (D)*** }]
[{ ***EVENT_NAME (D)*** }]
[{ ***NOEVENT_NAME*** }]

indicates if event data is to be reviewed. If /EVENT is not specified, console data is reviewed.

CONTEXT specifies that the context lines for each event be included in the output. Specify NOCONTEXT if you do not want the context lines included. The default is /EVENT=NOCONTEXT.

EVENT_NAME specifies that the event name, as specified by the system manager in the VCS scan files, should be included in the event line in the output. The default is /EVENT=EVENT_NAME.

/OUTPUT=file-name
indicates where the retrieved data is sent. If you do not enter the qualifier, or if you enter /OUTPUT without a file name, the output is sent to SYS$OUTPUT.

/PAGE
/NOPAGE (D)
indicates whether output from the VCS REVIEW command is displayed one screen at a time. Press Return to view the next page of output; press Ctrl/Z to quit the REVIEW display.

The default is /NOPAGE.

/SINCE=date-time
is the beginning point in the time period for which data is to be retrieved. The time and date must follow the VMS convention and in the format:

```
[dd-mmm-yyyy[:]][hh:mm:ss.cc]
```

This time must not predate the available log files and must not be in the future. It must always be an earlier time than the time specified by the /BEFORE qualifier. If /SINCE is not specified, then the starting point of the time period defaults to the current date and 00:00:00.00. If you specify a time without the date, it defaults to today at the specified time. If you specify a date without the time, it defaults to the specified date at 00:00:00.00.

/SYSTEM_NAME (D)
/NOSYSTEM_NAME
indicates whether the name of the serviced node associated with each line of data is to be included in the output. /NOSYSTEM_NAME is useful when reviewing data for a single node only.

The default is /SYSTEM_NAME.

/TIME_STAMP (D)
/NOTIME_STAMP
indicates whether the time stamp associated with each line of data is to be included in the output.

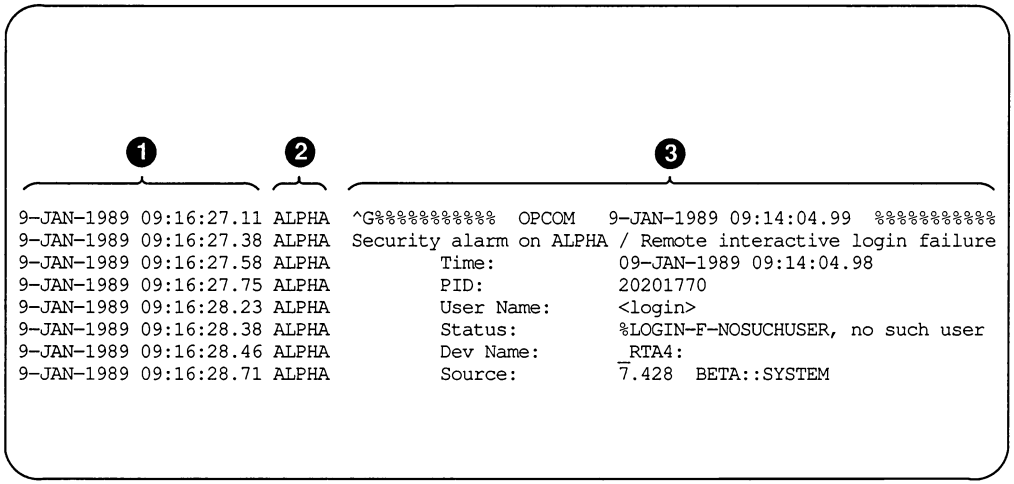
The default is /TIME_STAMP.

Description

Use the VCS REVIEW command to retrieve the log data in the VCS console or event log files for a specified time period. This command allows you to examine the log data that is no longer available in the log buffers of the Monitor Interface, as well as data that is current in the log buffers. The output of the VCS REVIEW command can be written to a file, a terminal or other output devices.

Retrieving console data is the default function. If you want to examine event data, you must use the /EVENT qualifier. Figure 5–2 describes the console data as displayed by the Review Interface.

Figure 5–2 Reviewed Text Anatomy


❶
❷
❸
9-JAN-1989 09:16:27.11 ALPHA ^G%%%%%%%%%% OPCOM 9-JAN-1989 09:14:04.99 %%%%%%%%%%
9-JAN-1989 09:16:27.38 ALPHA Security alarm on ALPHA / Remote interactive login failure
9-JAN-1989 09:16:27.58 ALPHA Time: 09-JAN-1989 09:14:04.98
9-JAN-1989 09:16:27.75 ALPHA PID: 20201770
9-JAN-1989 09:16:28.23 ALPHA User Name: <login>
9-JAN-1989 09:16:28.38 ALPHA Status: %LOGIN-F-NOSUCHUSER, no such user
9-JAN-1989 09:16:28.46 ALPHA Dev Name: _RTA4:
9-JAN-1989 09:16:28.71 ALPHA Source: 7.428 BETA::SYSTEM

MR-4079-RA

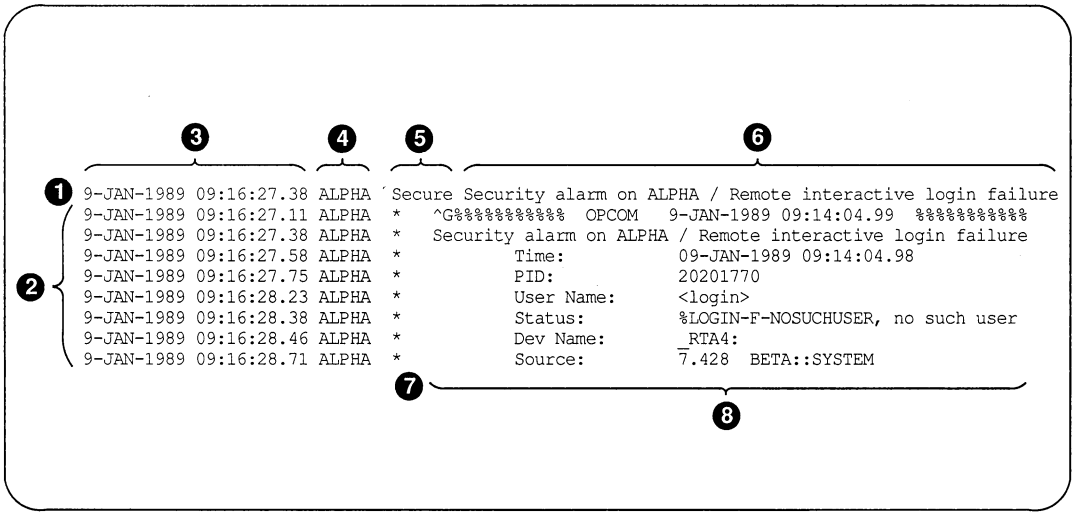
❶ The time stamp that is associated with the data line. If you want to omit the time stamp, specify /NOTIME_STAMP.

❷ The name of the serviced system from which the console data was received. If you want to omit the system name, specify /NOSYSTEM_NAME.

❸ The console data text as received from the serviced system.

The /EVENT qualifier allows you to retrieve the event data. Figure 5–3 describes the event data as displayed by the Review Interface.

Figure 5–3 Reviewed Event Anatomy


❸
❹
❺
❻
❶
❷
9-JAN-1989 09:16:27.38 ALPHA Secure Security alarm on ALPHA / Remote interactive login failure
9-JAN-1989 09:16:27.11 ALPHA * ^G%%%%%%%%%% OPCOM 9-JAN-1989 09:14:04.99 %%%%%%%%%%
9-JAN-1989 09:16:27.38 ALPHA * Security alarm on ALPHA / Remote interactive login failure
9-JAN-1989 09:16:27.58 ALPHA * Time: 09-JAN-1989 09:14:04.98
9-JAN-1989 09:16:27.75 ALPHA * PID: 20201770
9-JAN-1989 09:16:28.23 ALPHA * User Name: <login>
9-JAN-1989 09:16:28.38 ALPHA * Status: %LOGIN-F-NOSUCHUSER, no such user
9-JAN-1989 09:16:28.46 ALPHA * Dev Name: _RTA4:
9-JAN-1989 09:16:28.71 ALPHA * Source: 7.428 BETA::SYSTEM
❼
❽

MR-4080-RA

❶ The event information that was recorded in the event log file.

❷ The event context information, if any are available. The event context is displayed only when /EVENT=CONTEXT is specified.

❸ The time stamp that is associated with the data line. If you want to omit the time stamp, specify /NOTIME_STAMP.

❹ The name of the serviced system where the console data was received. If you want to omit the system name, specify /NOSYSTEM_NAME.

❺ The name of the event, as specified by the system manager in the scan files. If you want to omit the event name, specify /EVENT=NOEVENT_NAME.

❻ The console data text as received from the serviced system. It contains the text string that matches a predefined one as specified by the user in a scan event record. This text line may reflect a partial console text line since the Data Scanner may detect and notify the Event Logger of a match before the remainder of the console text line is received from the serviced node.

❼ The delimiter "*" indicates the event context data.

❽ The console data text as received from the serviced system.

Examples

1 `$ VCS REVIEW VAX_1,VAX_2 /NOTIME_STAMP /OUTPUT=REVIEW.LOG` Return

This command requests that the contents of the console log file for the current day be reviewed from the beginning of the day until the current time for the nodes VAX_1 and VAX_2. The output is written to a file called REVIEW.LOG.

2 `$ VCS REVIEW /EVENT=CONTEXT /SINCE=12:00:00 VCS$ALL_NODES` Return

This command requests that the contents of the event log file for the current day be reviewed from 12:00:00 until the current time for all the nodes in the current VCS configuration (VCS$ALL_ NODES). Also, the context lines for each event are to be included in the output. The output is written to SYS$OUTPUT.

3 `$ VCS REVIEW /SINCE=1-NOV-1988 /BEFORE=2-NOV-1988 NODE3,VCS` Return

This command requests that the contents of the console log file be reviewed for the entire day of November 1st, 1988, for NODE3 as well as any VCS data. The output is to be written to SYS$OUTPUT.

VCS SHUTDOWN

VCS SHUTDOWN

Shuts down all VCS components in an orderly manner.

Format

VCS SHUTDOWN

Description

As part of the shutdown, the current log files are closed, and all VCS service processes (for example, the I/O Data Logger, Event Logger, Data Scanner, and so on) are halted.

The Monitor Interfaces continue to run and receive a shutdown notification. All current data and log buffer information is still available. The Connect Interfaces are terminated and control is returned to either the Monitor Interfaces or DCL, whichever is the originator of that Connect Interface.

To restart VCS, issue the following command at DCL level:

```
$ @SYS$STARTUP:VCS$STARTUP Return
```

Example

```
$ VCS SHUTDOWN Return
```

This example shows the syntax for shutting down VCS at the DCL level. You can also shut down VCS within some of the interactive interfaces.

VCS START LOGGER

Starts the VCS logging processes.

Format

VCS START LOGGER *(/qualifier)*

Qualifiers

/CONSOLE (D)
Starts the I/O Data Logger process. /CONSOLE and /EVENT are mutually exclusive.

This is the default.

/EVENT
Starts the Event Logger process. /EVENT and /CONSOLE are mutually exclusive.

Description

The VCS START LOGGER command starts the I/O Data Logger and Event Logger. This command is normally invoked from the VCS$STARTUP.COM file as part of the system startup procedure. The I/O Data Logger process is the first VCS process to be started in the startup sequence. The Event Logger process is normally started immediately following the Data Scanner process.

Examples

1 `$ VCS START LOGGER /CONSOLE` Return

This command starts the I/O Data Logger process.

2 `$ VCS START LOGGER /EVENT` Return

This command starts the Event Logger process.

VCS START SCANNER

VCS START SCANNER

Starts the Data Scanner process.

Format

VCS START SCANNER

Description

This command is executed by the VCS startup command file, VCS$STARTUP.COM, as part of the system startup procedure. The Data Scanner process is normally started immediately following the I/O Data Logger process.

Example

```
$ VCS START SCANNER Return
```

This command starts the Data Scanner process.

6

Monitor Interface

This chapter consists of two parts:

1 An overview of the Monitor Interface. It discusses using the Monitor Interface in Connect and Monitor modes, as well as viewing log files and command processing.

2 A reference section listing each Monitor Interface command and its qualifiers.

6.1 Introduction

The Monitor Interface is a primary interface to VCS. It enables access to the console of any of the VCS serviced nodes. At the same time, the Monitor Interface provides information about the events which are occurring on the serviced nodes.

When you enter the Monitor Interface, VCS creates internal buffers for each of the serviced nodes specified in the VCS MONITOR command. Each buffer contains the console terminal output and any VCS events which occur for the serviced node. The buffers which contain the console data are referred to in this manual as log buffers or console log buffers. Initially, when VCS is started up on the system, each log buffer is empty until there is some activity on the console line. As the data on the console line accumulates, the log buffer will fill up. When the log buffer is full, the oldest data is dropped out of the buffer and the newer data continues to be collected in the buffer. All data is stored in the VCS console log files and the VCS event log files, so if data is not available in the Monitor Interface log buffers, it can be reviewed from the log files.

In the Monitor Interface, you can view each console log buffer through a window. The window shows you only the part of the log buffer that fits into the window, but you can move through the buffer using window scrolling commands. In this manual, the window into the log buffer is referred to as the log window or the console log window.

There are other buffers in the Monitor Interface, with a window into each. For instance, the Event Log buffer collects events as they occur, and you can view this buffer through the Event Log window. When you retrieve historical console data or events, the Monitor Interface switches you to the Review window, which is a window into the Review buffer. There is a Message buffer in the Monitor Interface, which accumulates all of the VCS and system messages sent to your process while you are running the Monitor Interface. You can view the Message buffer through the Message window. All of the windows you see in the Monitor Interface are windows into buffers.

The Monitor Interface functions in two modes: Connect and Monitor. These modes represent how the current terminal screen is managed.

6.1.1 Connect Mode

In Connect mode, using the Connect Interface, the terminal is connected by VCS to a system console line, similar to that provided by the DCL command SET HOST/DTE. VCS sends all characters, including screen control characters, from the application program, directly to your terminal. You enter Connect mode by issuing the CONNECT command. The result of Connect mode is that the VCS screen is under the control of any application being run on the system console. Screen-oriented applications, such as text editors, function correctly in Connect mode.

6.1.2 Monitor Mode

In Monitor mode, VCS completely controls the terminal screen. Programs that manipulate the terminal screen (for example, text editors) cannot control the screen. Screen control characters received by VCS are displayed as printable characters. The Monitor mode prevents an application program from interfering with the VCS display. Applications that require screen control must be used in Connect mode. For more legible output, set the serviced node's console port for hardcopy mode.

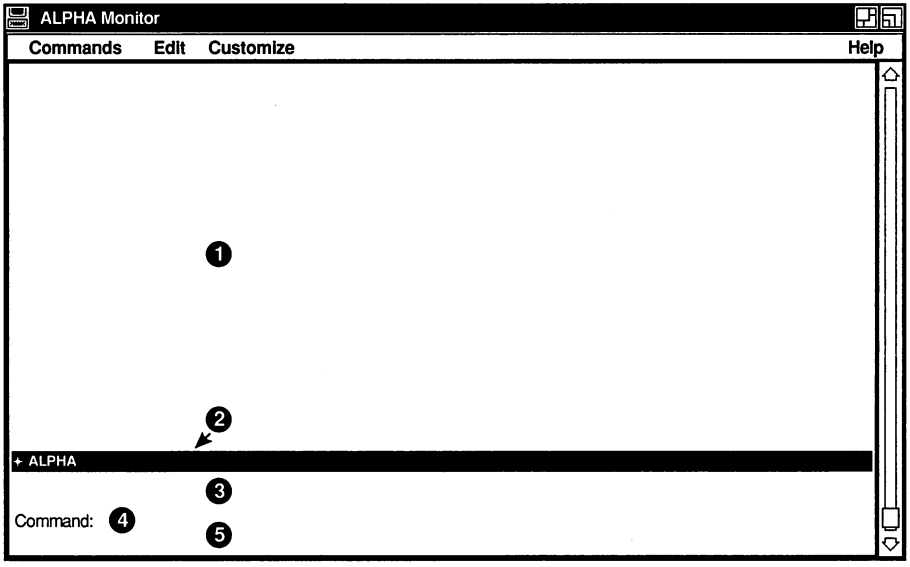
For example, on the serviced VAX/VMS system, issue the following DCL command:

```
$ SET TERMINAL OPA0: /NOSCOPE Return
```

Most of the VCS features are available in Monitor mode. In this chapter, all descriptions of commands apply only to Monitor mode. Anything applying to Connect mode is explicitly stated.

When you enter the DCL command VCS MONITOR to start the Monitor Interface session, the display fills your screen. The VCS screen display divides into the areas illustrated in Figure 6–1.

Figure 6–1 Monitor Interface Display with One Log Window



1. Log Window
2. Log Window Banner
3. Event Summary Window (unless you specified the /NOEVENT qualifier)
4. Command Line
5. Message Line

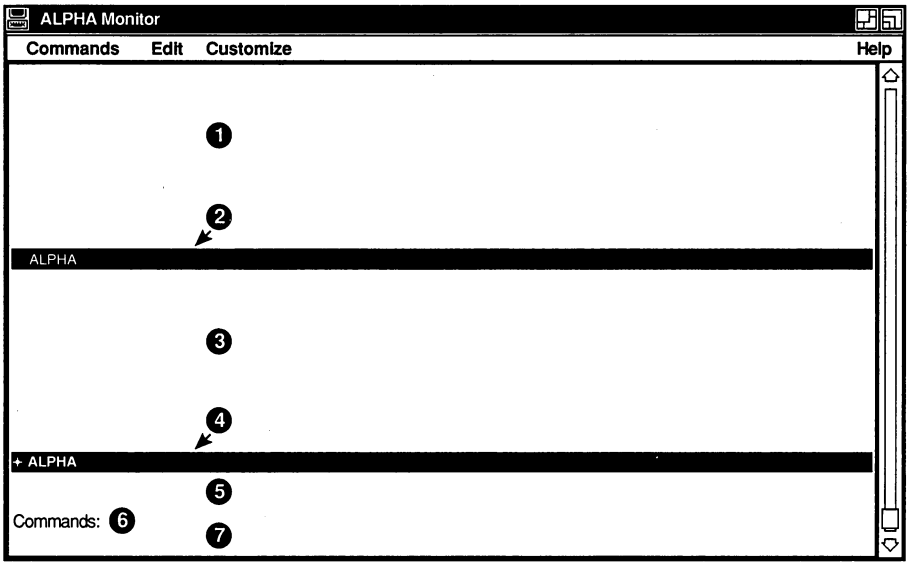
6.1.2.1 Log Window On a VT200-compatible terminal screen, the log window contains 20 lines consisting of 19 lines for the data and 1 line for the log window banner. The log window expands or shrinks to fit the available lines on other types of terminals. In addition, if you specify VCS MONITOR/NOEVENT, the log window contains 22 lines, 21 for data and 1 for the log window banner. The log window banner contains information about the contents of the log window or buffer you are viewing. The log window banner is in reverse video to provide a boundary between the windows on the screen. The banner contains the log window identifier, typically the node name and also the node herald from the configuration file. However, if you are not currently viewing a node, the identifier can be one of the following:

- EVENT LOG
- EVENT CONTEXT
- HELP
- MESSAGE
- REVIEW
- SHOW DIRECTORY
- SHOW KEYS
- SHOW MONITOR
- SHOW NODES

You can use the TWO_WINDOWS command to display two different logs or buffers. For example, you can display a log for a node in one window while you get help information in the other window. An asterisk (*) in the leftmost position of one log window banner indicates which log window is current. The banner text of the current window is also bolded. The current window receives the results of subsequent commands.

If you use the TWO_WINDOWS command to display two log windows on your screen, VCS divides the screen display into the areas illustrated in Figure 6–2:

Figure 6–2 Monitor Interface Display with Two Log Windows



1. Top Log Window
2. Top Log Window Banner
3. Bottom Log Window
4. Bottom Log Window Banner
5. Event Summary Window (unless you specified the /NOEVENTS qualifier)
6. Command Line
7. Message Line

On a VT200-compatible terminal, both the top and the bottom log windows contain 10 lines each, consisting of 9 for data and one for the log window banner. However, if you specify VCS MONITOR/NOEVENT, both log windows contain 11 lines each, 10 for data and one for the log window banner. The top log window banner contains information about the contents of the top log window, and the bottom log window banner contains information about the contents of the bottom log window.

The event summary window, command line, and message line are the same length and perform the same functions as they do in the one-window format.

When events and associated context information are displayed in the log window, they differ from the regular console data. Sections 6.1.2.1.1 and 6.1.2.1.2 discuss the concept of *current* event and how the events are displayed in the log window.

6.1.2.1.1 Event Log Window The VIEW/EVENT command displays the events in the log window. These events pertain only to the serviced nodes being monitored by the current Monitor Interface session and have accumulated since the session was started. As an event is detected, it is appended to the bottom of the event log buffer and is designated by the Monitor Interface as the *current* event. The Monitor Interface maintains a limit of approximately 300 of the latest events. When this limit has been reached, as new events are detected and added to the buffer, the oldest of the currently maintained events are discarded. You can use the REVIEW/EVENT command to retrieve the older events from the event log files.

The current event is displayed in the event log window as follows:

If the event priority is . . .	The event name is displayed in . . .	The event text is . . .
0 to 30	Plain text	Bolded
31 to 60	Reverse video	Bolded
61 to 100	Blinking	Bolded

Normally, the Monitor Interface highlights the newly arriving event, appends it to the event log buffer, and designates it the current event. However, you can also designate the current event with the following commands:

FIND
GOTO
NEXT EVENT
NEXT SCREEN
PREVIOUS EVENT
PREVIOUS SCREEN
SCROLL

Using these commands, you can navigate through the event log buffer. The event where your cursor is located is always the current event, which is highlighted by the Monitor Interface. If the current event is not the latest event at the bottom of the event log window, when a new event arrives, it is simply appended to the event log buffer and the event that is currently designated current, retains that status. However, if the current event is the latest event at the bottom of the event log window, when a new event arrives, it becomes the current event.

Thus you could use the above commands in the event log window as place holders or just prior to a VIEW/CONTEXT command. The VIEW/CONTEXT, NEXT EVENT, and PREVIOUS EVENT commands are keyed off the current event.

The designation of the current event remains under your exclusive control until you designate the latest event the current event. The GOTO LAST or sufficient NEXT SCREEN or SCROLL commands, for example, would make the latest event the current event. The current event designation remains with the most recent event, at the bottom of the event log buffer, until the next time you intervene.

6.1.2.1.2 Event Context Window The event context window displays the event context for the current event. The log window banner contains the window name EVENT CONTEXT and the console line with event text.

The context of an event is the number of contiguous console text lines. The event text may follow or fall anywhere within these text lines thereby giving more meaning to the *event* by placing it in *context*. The context may be displayed by the VIEW/CONTEXT, NEXT EVENT and PREVIOUS EVENT commands.

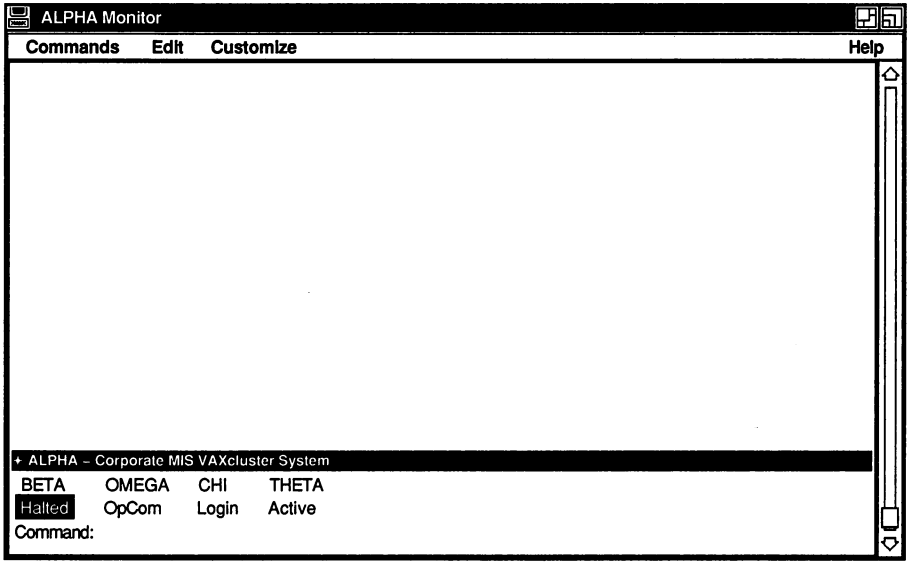
The context of an event is defined by the *context start* and *context count* fields of an EVENT record in the configuration file. See Chapters 3 and 9 for complete descriptions of these fields.

6.1.2.2 Event Summary Window The event summary window has two lines that provide event summary information about the nodes you are monitoring with VCS. The first line shows the node name, and the second line shows the event indicator for the node. The event indicators are ordered from highest to lowest priority and are placed from left (highest) to right (lowest) in the event summary window. The window contents are provided by the VCS Data Scanner. If you specify VCS MONITOR/NOEVENT, VCS does not include the event summary window in the Monitor Interface display. The system manager determines which events

VCS displays in the event summary window by supplying the appropriate information in the configuration file. See Chapters 3 and 4 for more information about scan event definitions.

Figure 6–3 shows a typical Monitor Interface display with an event summary window.

Figure 6–3 Event Summary Window



In Figure 6–3, the nodes BETA, OMEGA, and CHI are indicated to have events detected on their console lines. Whereas node THETA's console line is recognized as active but carries no particular information of interest. There may be additional nodes, but their absence from the event summary window implies that there has been no activity since they were last examined or since the Monitor Interface was started. When you examine the log window for a node which has an event in the event summary window, VCS clears that node and its event from the event summary window. If there are more indicators than there is room in the event summary window, then VCS displays a diamond character (◇) on the right edge of the event summary window. Use the SET WIDTH command to increase the width of the screen, which increases the width of the event summary window.

VCS displays only the highest severity event indicator for a node. For example, if a node receives both an **OpCom** and **Halted** indicator, only the highest priority indicator, **Halted**, is displayed in the event summary window.

VCS does not clear the indicators when the condition that caused them goes away. Instead, the indicators remain in the event summary window until you examine the log by executing the SELECT or VIEW command.

When VCS updates the event summary window, it sends a bell tone to the terminal. Use the SET BELL command to enable or disable the bell tone.

6.1.2.3 Command Line The command line is a one-line window that contains the VCS **Command:** prompt. The VCS commands you type are echoed here. When you view the Help buffer, the command line contains the prompt, **Topic?** or **Subtopic?**.

6.1.2.4 Message Line The message line is a one-line window that displays informational messages, such as error messages or system messages. You can use the VIEW/MESSAGE command to review previous messages that have been overwritten.

6.2 Command Processing in the Monitor Mode

Commands in VCS are processed in a manner similar to VMS DCL command processing. Line editing and recall features are available. The DCL similarities are outlined here and the differences are emphasized.

6.2.1 Line Editing Functions

Table 6–1 contains the commands and the character bindings of the DCL-adopted line editing functions. These predefined keys are active when you enter commands to the **Command:** prompt from the Monitor Interface, and also when you enter commands to the HELP facility of the Monitor Interface. While you are typing commands to a particular node (for example, after you have issued the CONNECT or SELECT command) your key definitions are dictated by the node to which you are connected. Therefore, the VCS command line editing functions will not work as described here while you are connected to the node.

Table 6–1 Command Line Editing Keys

Command	Keypad Characters
Execute	Return
Exit	Ctrl/Z
Previous command	↑ and Ctrl/B
Next command	↓
Move left	← and Ctrl/D
Move end line	Ctrl/E
Move right	→ and Ctrl/F
Move begin line	Ctrl/H, BS, and F12
Delete word to left of cursor	Ctrl/J, Linefeed, F13
Delete characters from beginning of line to cursor	Ctrl/U
Delete character to left of cursor	Delete
Discard current line and delete data in the type-ahead buffer	Ctrl/X
Quote	Ctrl/V
Toggle between overstrike and insert mode	Ctrl/A or F14

Note that the retype function (Ctrl/R) from DCL is not included.

The functions in Table 6–1 (excluding EXIT) cannot be bound to other characters because they do not exist as separate commands. For example, if you bind Ctrl/V to some other command, you cannot bind QUOTE again to any character.

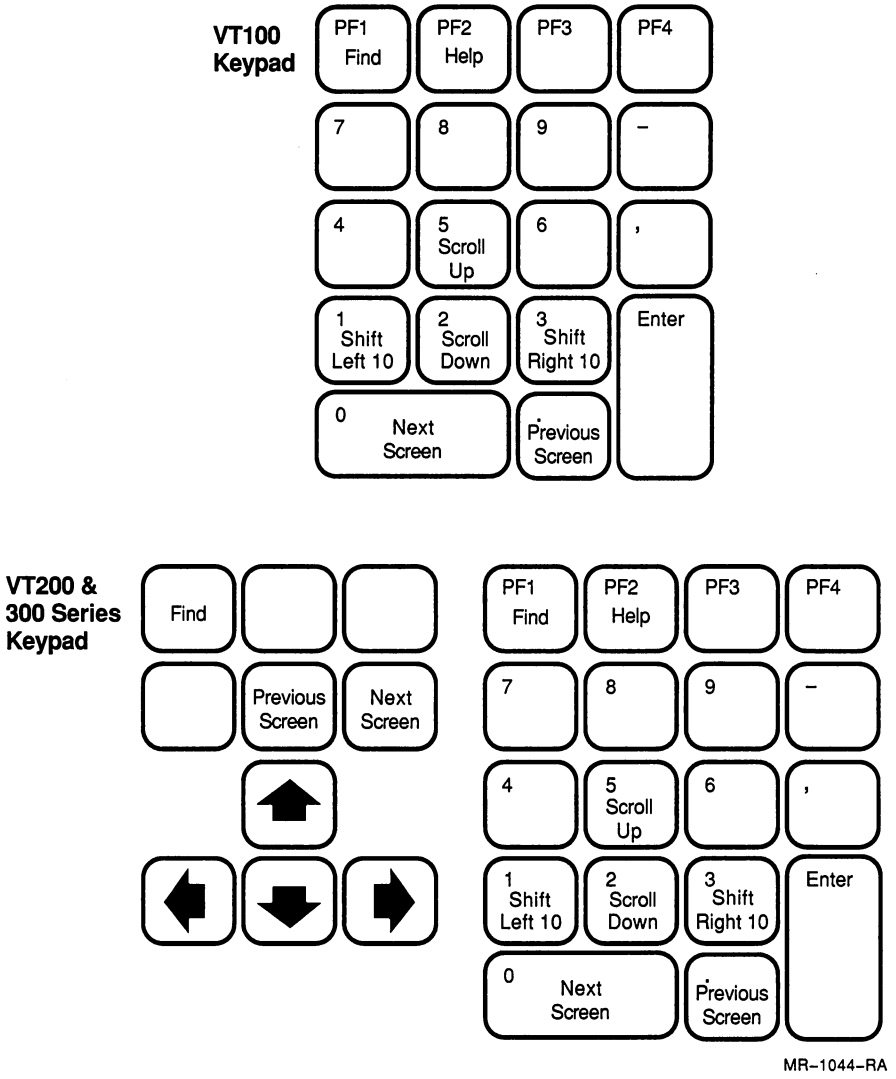
6.2.2 Key Definitions

VCS enables you to execute some Monitor Interface commands by pressing a predefined key instead of entering the command on the command line. You also can define keys with the DEFINE/KEY and SET SHIFT_KEY commands. You can unbind key definitions by using the UNDEFINE/KEY command. You can save the key definitions with the SAVE command. You can view the key definitions with the SHOW KEYS command.

The process update function (Ctrl/T) has been bound to the VCS command SHOW TIME and thus behaves differently from its DCL counterpart.

Figure 6–4 shows you the keypad definitions for both the VT100 and the VT200/300-series terminals.

Figure 6–4 Window Control Keys



MR–1044–RA

6.2.3 VAXstation Mouse

You can use the VAXstation mouse to control the Monitor Interface window display.

The Do key toggles the enabling and disabling of the mouse. By default, the leftmost button of the mouse is MB1, the middle button is MB2, and the rightmost button is MB3.

To select a window, move the pointer to within the bounds of the window you want, then press and release the required button. This action is called *clicking*.

Most of the window display functions require you to move the mouse while holding a mouse button down and then release the button when the pointer is where you want it. This action is called *dragging*.

6.2.4 Node Names

Many commands require a node name. Node names are ASCII strings of six or less characters in length and can be abbreviated to their minimum unique character string. You can define a logical name to represent one or more node names. These logical names can be used anywhere a node name is required. VCS will always try first to translate a node name as a logical name. Logical names cannot be abbreviated. See Section 5.1 for more information on how to specify node names to VCS commands.

6.2.5 Text Strings Contained in Command Lines

A text string is a set of characters which may be enclosed in quotes ("*string*"). A string is designated in command descriptions as *string*. Generally, quotes are not necessary because the string is usually included as the last argument of any command. If a string is enclosed in quotes, the first non-space character must be the quote (") character. Otherwise, the quote character is included within the string. This is different from standard DCL, which requires quotes if the string includes spaces. Therefore, it is not necessary for you to enclose all strings using the OUTPUT command in quotes. Any lowercase unquoted string arguments to the OUTPUT command are converted to uppercase by the DCL routines. You must enclose the string in quotes or use the CONNECT or SELECT command to pass lowercase characters onto the node's console line.

6.2.6 Specifying Control Characters

Some commands require binding a single key to a particular function such as break or escape characters. VCS specifies these characters with a keyword. VCS uses the keyword **CTRL_A_KEY** to specify Ctrl/A, **CTRL_B_KEY** to specify Ctrl/B, and so on. If a control character has been assigned a function, the control character cannot be typed without invoking the function. To type a control character literally, the character must be preceded by the line-editing quote character Ctrl/V.

6.2.7 Date and Time Arguments

Specify date and time arguments in the standard DCL format. VCS accepts time arguments in absolute and delta formats. The format for absolute time is [*dd-mmm-yyyy*[:]][*hh:mm:ss.cc*]

You can specify the date and time either individually or together. If you specify both, you must separate them with a colon (:).

Delta time is an offset from the current time to a time in the future. When using delta time, you can include only the time; you cannot include a date. The format for delta time is [*hh:mm:ss.cc*]

Delta time must begin with the number of hours and not a colon, even if the number of hours is equal to zero.

6.3 Monitor Interface Commands

The Monitor Interface commands, described in the remainder of this chapter, are listed in Table 6–2.

Table 6–2 Monitor Interface Commands

@	PREVIOUS EVENT	SET TIME_STAMP
CONNECT	PREVIOUS SCREEN	SET WIDTH
CYCLE	RECONFIGURE	SHIFT
DEFINE/KEY	REFRESH	SHOW DIRECTORY
DISABLE	REVIEW	SHOW KEYS
ENABLE	SAVE	SHOW MONITOR
EXIT	SCROLL	SHOW NODES
FIND	SELECT	SHOW TIME
GOTO	SET BELL	SHUTDOWN
HELP	SET BREAK	TWO_WINDOWS
NEXT EVENT	SET ESCAPE	UNDEFINE/KEY
NEXT SCREEN	SET LOCK	VIEW
ONE_WINDOW	SET MOUSE	WAIT
OTHER_WINDOW	SET PRIVILEGES	WRITE
OUTPUT	SET SHIFT KEY	

@ (Execute Command File)

Executes a Monitor Interface command file.

Format

@ *file-name*

Parameters

file-name
Specifies the Monitor Interface command file to be executed. If you do not specify a file type, the Monitor Interface uses the default file type of COM. No wildcard characters are allowed in the file specification.

Description

When you specify the @ command to execute a command file at the **Command:** prompt, the Monitor Interface switches the input command stream source from the terminal to the specified file. The file should contain commands as they would be typed at the **Command:** prompt. Commands that need arguments must contain the arguments in the command line.

A command file can also contain the @ command to execute another command file. The maximum command level you can achieve by nesting command files is ten, including the top-level command file.

The WAIT command is an ideal tool to set the tempo for your command file execution. When you compose the contents of a command file that contains several OUTPUT commands, you may want to insert the WAIT commands between the appropriate commands to allow the serviced system time to process the text strings. For example, the HSC device may take up to a second to respond to the Ctrl/C entered at its console. The Monitor Interface may execute a series of OUTPUT commands so rapidly that the HSC console port is overrun.

You may want to create such a series of OUTPUT commands as follows:

```
OUTPUT/NOTERMINATE HSC001 Ctrl/V Ctrl/C
WAIT 00:00:02.00
OUTPUT HSC001 "SHOW DISKS"
```

The Monitor Interface ignores the CYCLE command in a command file.

Example

```
$  CREATE STATUS.COM Return
OUTPUT ALL_VAXES SHOW TIME Return
OUTPUT ALL_VAXES SHOW SYSTEM Return
OUTPUT ALL_VAXES SHOW USER Return
Ctrl/Z
$  VCS MONITOR ALL_VAXES Return
Command:  @STATUS.COM Return
```

This example shows a command file, named STATUS.COM, that sends the DCL commands to all VAX systems to show time, system processes, and login users.

CONNECT

Activates the Connect Interface to connect your terminal to the node you specify and allows you to communicate with that system as if you were on a local console terminal.

Format

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

The default node name is the one that you are viewing in the *current* window.

Qualifiers

/ACTIVE
indicates the first node, leftmost one, listed in the event summary window as the node you want to communicate with.

/BREAK_KEY=key-name
/NOBREAK_KEY
specifies a control character that causes a Break signal to be generated on the console line to the connected node. The key name can be a control character in the form CTRL_*x*_KEY, where *x* is an alphabetic character A through Z; BS_KEY; or LF_KEY.

/NOBREAK_KEY indicates there is no way to generate a Break signal to the connected node.

The default break character is the one defined by the last SET BREAK command. If no prior SET BREAK command was entered, there is no break key defined. Use the SHOW MONITOR command to determine the currently defined break character.

/CONTINUE (D)
/NOCONTINUE
indicates whether the interface continues to accept input from SYS$COMMAND after the contents of the files you specified with the /INPUT qualifier are successfully transmitted. This qualifier is ignored if you do not specify the /INPUT qualifier.

Use /NOCONTINUE to issue an implicit escape key immediately after the contents of the last file is transmitted successfully.

The default is /CONTINUE.

/ESCAPE_KEY=key-name
specifies a control character or key that enables you to return to the Monitor Interface. The key-name can be either a control character in the form CTRL_*x*_KEY, where *x* is an alphabetic character A through Z; BS_KEY; or LF_KEY.

If you do not specify this qualifier, VCS uses the escape character defined by the last SET ESCAPE command. If no prior SET ESCAPE command was entered, the default escape key is Ctrl/G.

/INPUT=file-name(, . . .)
is an ASCII text file whose content is to be transmitted. When you specify a list of files, separate the names with commas. If any of the files in the list do not exist, none of them will be transmitted.

If a file contains a control character that is identical to the one you specified with the /ESCAPE_KEY qualifier, VCS sends the character to the node as a regular control character. Typing the escape character at the controlling terminal while the files are

being sent will halt the transmission temporarily. VCS prompts you with the following message:

```
Do you want to exit (Yes/No)?
```

If you answer no, the transmission continues.

If a file contains a control character that is identical to the one you specified with the /BREAK_KEY qualifier, VCS translates the character into a Break signal on the console line.

/LOCK
/NOLOCK (D)
indicates whether you intend to reserve the exclusive input rights to the connected node's console line. If you do, other users cannot send data to the node, but they can still examine log information and view your current activity. The node remains locked until you disconnect from the node by pressing the escape character. /NOLOCK means that you share input rights with other users on the specified console line. Any other operator would be able to enter data simultaneously on the console line.

The default is /NOLOCK.

/NEXT
Connects you to the subsequent node in the list you specified with the VCS MONITOR command.

/RATE= { ***FAST*** | ***MEDIUM*** | ***SLOW (D)*** }
is the rate at which characters are sent from the /INPUT file to the nodes. FAST indicates that the contents of the files will be transmitted as fast as possible to the node. MEDIUM indicates that the contents of the files will be transmitted to the node at about 8 to 10 characters per second. SLOW indicates that the contents of the files will be transmitted to the node at about 1 to 3 characters per second.

The /RATE qualifier does not change your baud rate. The /RATE qualifier is ignored if you do not specify the /INPUT qualifier.

The default rate is SLOW.

CONNECT

Description

The CONNECT command connects your terminal to the node you specify. The node controls your full terminal screen. You may want to use this command to run a program that requires full-screen control, or when you want the full terminal screen for your work.

When you type the CONNECT command, VCS displays a message in one of the two formats similar to the following messages:

```
%VCS-I-SHOWNBKESC, No break is set, CTRL/G is escape
%VCS-I-SHOWBRKESC, CTRL/B is break, CTRL/G is escape
```

The %VCS-I-SHOWNBKESC message means that you have not defined a break character and the default escape character is Ctrl/G. If you use the SET ESCAPE or the /ESCAPE_KEY qualifier with the CONNECT command to define your own escape character, VCS displays the character you defined. If you use the SET BREAK command or the /BREAK_KEY qualifier with the CONNECT command to define a break character, VCS displays the %VCS-I-SHOWBRKESC message which includes both the break and escape characters.

Once you have a connection to the specified node through the console line, characters (excluding the substitute break and escape characters, Ctrl/S and Ctrl/Q), are transmitted directly to the node. Output from the node is sent to your terminal untranslated. This continues until you type one of the designated control characters and you are prompted to verify the action with one of the following:

```
Do you want to exit (Yes/No)?
Do you want to transmit a break (Yes/No)?
```

If you answer no, the actual control character entered is transmitted to the console. If you answer yes, the requested action occurs.

If you define both the substitute break and escape characters to be the same, the escape action takes precedence over the break action. For example, if you define both characters to be CTRL_G_ KEY, when you press Ctrl/G, VCS asks whether you intend to exit the Connect Interface. However, during the session, you will not be able to transmit any Break signal using Ctrl/G.

Note *Any text you type on a terminal connected to a console line is echoed on all terminals connected to the same console line with the Monitor or Connect Interfaces.*

Example

```
Command:  CONNECT ALPHA Return
%VCS-I-SHOWNBKESC, No break is set, CTRL/G is escape
Return
$
```

This example shows you how to connect to the node named ALPHA. When you connect to the node, the Connect Interface displays a message that shows your current break character and escape character. You can then enter commands to ALPHA.

After you finish your work on ALPHA, press the escape character Ctrl/G to return to the Monitor Interface display. The Connect Interface prompts you to confirm that you want to leave the connected node. If your answer is no, the control-character is passed on to the node unchanged. If your answer is yes, the Connect Interface disconnects from the node and returns you to the Monitor Interface.

CYCLE

Continually displays the contents of the log buffers, one after another, automatically.

Format

CYCLE *(/qualifier)*

Qualifiers

/INTERVAL=delta-time
is the time interval that a node log is maintained in the current window before the next one is displayed. The minimum interval is one second.

The default interval is 10 seconds.

Description

The CYCLE command enables you to view all logs, one at a time, on the current window at a set rate of speed. The CYCLE command has the same affect as if you issued a series of VIEW/NEXT and WAIT commands.

While the command is active, the Monitor Interface ignores all user input from the keyboard. To stop the cycle and return control to the terminal, enter the escape character. The Monitor Interface ignores the CYCLE command in an indirect command file.

Example

```
Command:  CYCLE/INTERVAL=00:00:40 Return
```

In the sample command line, each node log is maintained in the current window for forty seconds.

DEFINE/KEY

Binds a text string to a key for the current Monitor Interface session.

Format

DEFINE /KEY *(/qualifier . . .) key-name string*

Parameters

key-name
is the name of the key or control character to which you assign a string. Use the key names specified in Table 6–3 to define control keys.

Table 6–3 Control Key Names

CTRL_A_KEY	CTRL_H_KEY	CTRL_O_KEY	CTRL_V_KEY
CTRL_B_KEY	CTRL_I_KEY	CTRL_P_KEY	CTRL_W_KEY
CTRL_C_KEY	CTRL_J_KEY	CTRL_Q_KEY	CTRL_X_KEY
CTRL_D_KEY	CTRL_K_KEY	CTRL_R_KEY	CTRL_Y_KEY
CTRL_E_KEY	CTRL_L_KEY	CTRL_S_KEY	CTRL_Z_KEY
CTRL_F_KEY	CTRL_M_KEY	CTRL_T_KEY	LF_KEY
CTRL_G_KEY	CTRL_N_KEY	CTRL_U_KEY	BS_KEY

You cannot define the Return key and all printable alphanumeric characters. Also, it is recommended that you do not redefine the control characters specified in Table 6–4. If you do redefine one of these control characters, you cannot access the function originally associated with it during the current Monitor Interface session.

Table 6–4 Reserved Control Key Names

CTRL_A_KEY	CTRL_E_KEY	CTRL_J_KEY	CTRL_U_KEY
CTRL_B_KEY	CTRL_F_KEY	CTRL_Q_KEY	CTRL_V_KEY
CTRL_D_KEY	CTRL_H_KEY	CTRL_S_KEY	CTRL_X_KEY

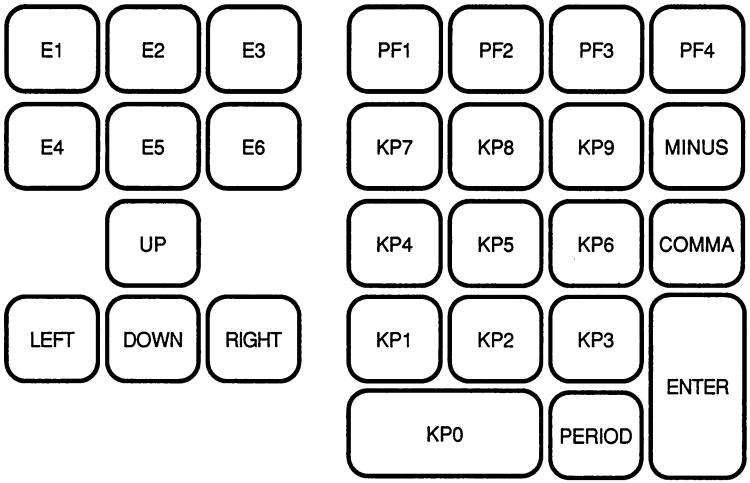
Use the key names specified in Table 6–5 to define editing, keypad, or function keys.

Table 6–5 Function and Keypad Key Names

PF1	PF2	PF3	PF4	KP0	KP1
KP2	KP3	KP4	KP5	KP6	KP7
KP8	KP9	PERIOD	COMMA	MINUS	ENTER
UP	DOWN	LEFT	RIGHT	E1	E2
E3	E4	E5	E6	HELP	DO
F6	F7	F8	F9	F10	F11
F12	F13	F14	F15	F16	F17
F18	F19	F20			

The keys in Table 6–5 are located on the editing and application keypads as shown in Figure 6–5.

Figure 6–5 Keypad Layout



string
is a sequence of keystrokes, typically a command and its parameters. Although quotation marks are not required as string delimiters, you must include them if you want to retain extra spaces and lowercase letters.

Qualifiers

/KEY
is required with the DEFINE command to assign a command or command string to a key.

/SHIFT_KEY
/NOSHIFT_KEY (D)
indicates if the key, when executed, is to be preceded by the *shift* key that is defined by the SET SHIFT_KEY command. /SHIFT_KEY enables you to assign two command strings to one key: one is used when you press the key by itself, and the other is used when you press the key after the defined *shift* key. You define the *shift* key with the SET SHIFT_KEY command.

The default is /NOSHIFT_KEY.

/TERMINATE
/NOTERMINATE (D)
indicates if a carriage-return character is to be included as part of the string you assign to a key. Thus, you can execute the command simply by pressing the defined key. If you specify /NOTERMINATE, the carriage-return character is not included as part of the string. Pressing the defined key causes the Monitor Interface to insert the associated string at the current location on the command line. To complete and execute the command, you must then press Return.

The default is /NOTERMINATE.

Description

You can save time by assigning a frequently used command or string to a key. You can then issue the command by pressing one key, instead of typing the full command name.

Quotation marks are not required as string delimiters. For example, either the string **SELECT ALPHA** or the string **"SELECT ALPHA"** prints as **SELECT ALPHA** on your terminal. You must use quotation marks if you want to retain spaces and lowercase letters. For example, **select alpha** prints as **SELECT ALPHA** and **"select alpha"** prints as **select alpha**. Extra spaces in an unquoted string are compressed into a single space.

For a quotation mark to be the first printed character in the string, you must enclose the string in a second set of quotation marks. For example, **""SELECT ALPHA""** prints as **"SELECT ALPHA"**.

You can redefine keys that were defined previously. The Monitor Interface informs you of the overridden definition with a message similar to either one of the following:

```
%VCS-I-MONS_USERKEYWAS, KP9 definition was "SHOW MONITOR"
%VCS-I-MONS_VCSKEYWAS, KP0 function was "Next screen"
```

The %VCS-I-MONS_USERKEYWAS message indicates that a user-defined key was redefined. It displays the previously defined command string that was bound to the key. The %VCS-I-MONS_VCSKEYWAS message indicates that a VCS predefined key was redefined. It displays the original VCS function that was bound to the key.

The key definitions remain in effect for your current session only, unless you preserve the key definitions with the SAVE command. Once you SAVE the key definitions, you can reactivate them by specifying the VCS MONITOR/SECTION command.

Use the UNDEFINE/KEY command to remove a key definition. The undefined state of a key also remains in effect for your current session only. That is, to permanently have a key undefined, you also need to SAVE that state. Otherwise, the next time you invoke the Monitor Interface, the old definition remains the same.

Use the SHOW KEYS command to display the current key definitions.

Example

```
Command:  DEFINE/KEY KP1 CONNECT [Return]
Command:  [KP1]
Command: CONNECT  ALPHA [Return]
```

This example shows how to define key KP1 so that, when you press KP1, the text CONNECT prints on the command line. You can then complete the command by entering the node name ALPHA followed by Return.

DISABLE

Temporarily stops the services, including event and console data logging and communications, between VCS and the specified node.

Format

DISABLE *node-name(, . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

Description

You must issue the SET PRIVILEGES command before you can enter the DISABLE command. Typically, you disable a node from VCS only in special circumstances. For example, when you want to shut down a node temporarily, use the DISABLE command to discontinue VCS services for the node. If you plan to remove a node permanently, you should use the Configuration Editor to remove the node from the configuration file, and use the RECONFIGURE command to reconfigure the running VCS system.

Before you issue the DISABLE command, you may want to ensure that you do not lose the information transmitted from the node over the console line. You also may need to control the node from an alternate console terminal.

The display generated by the SHOW NODES command indicates whether a node is enabled or disabled. Also, the message **[System is Disabled]** appears at the end of the log buffer of the node.

Use the ENABLE command to resume VCS services for a node.

Example

```
Command:  SET PRIVILEGES Return
Command:  DISABLE AUTO Return
Command:  SHOW NODES Return
```

This example shows you how to disable the node AUTO from VCS processing. After you issue the DISABLE command, you can enter the SHOW NODES command to verify that you disabled the node.

```
    NODE      DEVICE    ENABLE   LOG
NO  NAME      NAME      STATUS   STATUS   LOCK STATUS     SCAN PROFILE

 1  AUTO      LTA917    No       Yes      Off             VMS9000
 2  ACCNT     LTA511    Yes      Yes      Off             VMS6400
```

ENABLE

Counteracts a previously issued DISABLE command, resumes logging of console and event data for the specified node, and allows user communications with the node through the VCS interfaces.

Format

ENABLE *node-name(, . . .)*

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

Description

You must issue the SET PRIVILEGES command before you can enter the ENABLE command. If you switched console line monitoring from VCS to an alternate terminal before issuing the DISABLE command, you will need to establish the proper hardware connections before issuing the ENABLE command.

If you entered the DISABLE command during a previous VCS session and you want to enable the node during the current session, you must specify the name of the node as one of the parameters for the VCS MONITOR command. For example, if you disabled node AUTO during an earlier Monitor Interface session and then exit, you must include AUTO as one of the node names for the VCS MONITOR command to enable VCS processing for the node during the current session.

Example

```
Command:  SET PRIVILEGES [Return]
Command:  ENABLE AUTO [Return]
Command:  SHOW NODES [Return]
```

This example shows you how to reconnect the previously disabled node AUTO to VCS. After you issue the ENABLE command, you can enter the SHOW NODES command to verify that VCS processing has resumed for the node.

```
    NODE     DEVICE   ENABLE   LOG
NO  NAME     NAME     STATUS   STATUS   LOCK STATUS    SCAN PROFILE

 1  AUTO     LTA917   Yes      Yes      Off            VMS9000
 2  ACCNT    LTA511   Yes      Yes      Off            VMS6400
```

EXIT

Terminates a Monitor Interface session.

Format

EXIT

Description

Use the EXIT command to end the session and return to the DCL command level. You can also issue the EXIT command by pressing Ctrl/Z. To enter a Monitor Interface session again, use the VCS MONITOR command at the DCL command level.

If you locked any nodes during the session, they are released and others can now access and send data to those systems.

Example

```
Command:  EXIT  Return
$
```

When you enter EXIT on the VCS command line, the Monitor Interface session ends, and VCS returns you to the DCL level.

FIND

Searches the current Monitor Interface buffer for a specified text string.

Format

FIND *(/qualifier) (string)*

Parameters

string
is a text string to be searched. If the searched text string contains spaces, it must be quoted to preserve them.

The default is the string you previously searched for in this session.

Qualifiers

/FORWARD (D)
indicates the forward direction in time. That is, search from the current location in the buffer to the most current information for the string you specify.

The default direction is /FORWARD, the first time you specify the FIND command in the current Monitor Interface session.

/REVERSE
indicates the backward direction in time. That is, search from the current location in the buffer to the least current information, for the string you specify.

FIND

Description

Use the FIND command to search through a buffer for the occurrence of a string, such as a time stamp or an event. If you do not specify /FORWARD or /REVERSE, VCS searches in the direction you last specified with a FIND command. If this is the first FIND executed in the session, the default direction is forward.

If you do not specify a string to be found, VCS searches for the last string that you specified with the FIND command. For example, if you type FIND/REVERSE OPCOM, VCS searches in a reverse direction for the string **OPCOM**. If you issue FIND again, VCS continues searching in the reverse direction for a previous occurrence of the string **OPCOM**.

When the FIND command locates a string, it highlights the string in reverse video on your screen. If the FIND command cannot locate the string, it displays a message that it failed to the string.

After you enter a complete FIND command, you can request searches for the same string by pressing the FIND key repeatedly. You can execute the FIND command by pressing Find on the VT200 editing keypad or PF1 on the VT100 or VT200 numeric keypad.

Examples

1
```
Command:  VIEW JUNO [Return]
Command:  GOTO FIRST [Return]
Command:  FIND OPCOM [Return]
```

The VIEW command displays current log information for node JUNO. GOTO FIRST moves the current window to the oldest information in the log buffer. The FIND command then searches forward in time through this buffer for the string **OPCOM**.

2
```
Command:  REVIEW MARS [Return]
Command:  FIND/REVERSE BACKUP [Return]
```

The REVIEW command displays console log information from the log file for node MARS. The FIND/REVERSE command searches the REVIEW buffer for the string BACKUP. The search begins at your current location and continues toward the oldest information in the log.

3
```
Command:  VIEW/EVENT [Return]
Command:  FIND/REVERSE OPCOM [Return]
```

The VIEW/EVENT command displays events from all currently monitored nodes since the Monitor Interface was started. The FIND/REVERSE command searches backward in time to locate the text string **OPCOM**. The line of text on which the string **OPCOM** is found is designated to be the current event.

GOTO

Moves the log window to the top or bottom of the current buffer.

Format

GOTO { FIRST / LAST }

Parameters

FIRST
displays, in the current window, the oldest information for the buffer associated with the window.

LAST
displays, in the current window, the newest information for the buffer associated with the window.

Description

Use the GOTO command to move the current window to the start or end of the buffer. An asterisk (*) in the log window banner indicates the current window. If you have two windows, you can use the OTHER_WINDOW command to make the other window the current window.

Examples

1
```
Command:  VIEW THETA Return
Command:  GOTO FIRST Return
```

The VIEW command displays current log information for THETA. When you enter GOTO FIRST, VCS displays the oldest log information for THETA.

2
```
Command:  REVIEW ORANGE Return
Command:  GOTO LAST Return
```

The REVIEW command displays console log information for the node named ORANGE. When you enter GOTO LAST, VCS displays the newest console log information for ORANGE.

HELP

Invokes the online help facility.

Format

HELP *(topic)*

Parameters

topic
The name of the command for which you requested help. If you omit the topic, VCS displays a list of available topics.

Description

The HELP facility displays information about requested VCS topics. From the Monitor Interface, you can display a list of topics by typing HELP and pressing Return. VCS responds by displaying a brief description of how to use HELP, followed by a list of topics for which help is available. The prompt is changed to **Topic?**. In response to the prompt, you can do the following:

- Press the following keys to scroll through the help buffer:
  - Next Screen or keypad 0
  - Prev Screen or keypad period
  - Keypad 5 to scroll up
  - Keypad 2 to scroll down
- Enter one of the topic names from the list.
- Enter HELP DESCRIPTION for more information about using the help facility.
- Press Return one or more times, or press Ctrl/Z once to exit from the online help facility.
- Enter a question mark (?) to see a list of topics.

- Enter an asterisk (*) to display all available information in the HELP file.
- Enter a topic immediately followed by an ellipsis (. . .) to display all of the information on a topic. For example, CONNECT . . . displays all of the information available for the CONNECT command.

You can abbreviate any topic name, although ambiguous abbreviations result in all matches being displayed.

When you type the HELP command, VCS clears the help buffer and then places the information that you request in the help buffer. VCS displays the HELP buffer on your screen. The log window banner displays **Help** to show that you are viewing the help buffer. The command line prompt, **Command:** changes to **Topic?**. When you request information on a topic, the **Topic?** prompt then changes to *topic* **Subtopic?**, for example, **CONNECT Subtopic?**.

Proceed forward and back in the help buffer with KP5, KP2, Next Screen or KP0, and Prev Screen or KP. keys. Use Return to return to a previous topic level. Eventually, pressing Return brings you back to the top level of the help prompt and subsequently back to the Monitor Interface **Command:** prompt.

When you exit HELP, the help buffer retains the current contents until you issue another HELP command. You can recall the current help text by giving the command VIEW/HELP. The current contents of the help buffer remain on your screen until you issue another command such as SELECT, VIEW, REVIEW, CONNECT, or a SHOW command.

Examples

1 Command: HELP HELP DESCRIPTION Return

When you enter this command, the Monitor Interface displays a description about how to use the online help facility.

2
```
Command:  VIEW SATURN Return
Command:  TWO WINDOWS Return
Command:  HELP CONNECT Return
```

This example shows you how to view current log information for the node SATURN in one window while you display help information for the CONNECT command in the other window.

NEXT EVENT

Navigates forward in time the list of events in the Event Log or Event Context window.

Format

NEXT EVENT

Description

The NEXT EVENT command is valid only if the current window is either the Event Log window or the Event Context window. If you enter the NEXT EVENT command in any other window, an error message is issued. Event order is determined by the order of the events in the Event Log buffer. The next newer event is the next event toward the bottom of the Event Log buffer, from the current event.

If the current window is the Event Log window, the NEXT EVENT command makes the next newer event the current event. The current event is highlighted according to its priority, as described in Section 6.1.2.1.1. If the current window is the Event Context window, the NEXT EVENT command displays the context of the next newer event, while at the same time making that event the current event.

If no newer event exists, a warning is issued and the current event does not change.

Examples

1
```
Command:  VIEW/EVENT  Return
Command:  GOTO FIRST  Return
Command:  NEXT EVENT  Return
```

This sequence of commands enables you to view the Event Log window, then go to the top of that window, and make the next newer event the current event. As you issue each of these commands, notice that the current event is changed. The highlighting of the event text shows you which event is current.

2
```
Command:  VIEW/CONTEXT  Return
Command:  NEXT EVENT  Return
```

This sequence of commands enables you to view the Event Context window and then view the context of the next newer event after the current event. If no newer event exists, a warning is generated.

3
```
Command:  VIEW/EVENT  Return
Command:  TWO_WINDOWS  Return
Command:  VIEW/CONTEXT  Return
Command:  NEXT EVENT  Return
```

This sequence of commands enables you to view the Event Log window, then split the screen into two windows and view the Event Context window in the current window of the two windows. Issuing the NEXT EVENT command then causes the next newer event to become the current event. You will see it become highlighted in the Event Log window. The same NEXT EVENT command also causes the Event Context window (the current window) to display the context of the event which has just become the current event.

NEXT SCREEN

Moves the current window forward in time by one window over the current buffer information so that the next window of data is visible.

Format

NEXT SCREEN

Description

The NEXT SCREEN command displays data one window at a time, from the current position in the buffer to the next window of information. You can execute the NEXT SCREEN command by pressing Next Screen on the VT200-series editing keypad or KP0 on the VT100 numeric keypad.

If you specify NEXT, the command defaults to NEXT SCREEN.

Example

```
Command:  REVIEW CRYSTL [Return]
Command:  GOTO FIRST [Return]
Command:  NEXT SCREEN [Return]
```

This command displays historical log information for the node named CRYSTL. Entering a GOTO FIRST brings the display back to the beginning of the buffer. When you type NEXT SCREEN, the display moves the current window forward by one window.

ONE_WINDOW

If there are two windows, the command removes the window not currently selected and enlarges the current window to fill the available log window space.

Format

ONE_WINDOW

Description

Use the ONE_WINDOW command to change your screen display from two windows to one window. The ONE_WINDOW command retains and enlarges the current log window on the screen and removes the other window from the screen. An asterisk (*) in the log window banner indicates the current window.

Example

```
Command:  ONE_WINDOW Return
```

This example shows how to change the screen from one window to two windows.

OTHER_WINDOW

If there are two windows, the OTHER_WINDOW command makes the non-current window current.

Format

OTHER_WINDOW

Description

The TWO_WINDOWS command divides the available log window space into two windows of equal size. The *current* window is flagged by an asterisk (*) before the system name in the bolded log window banner. The OTHER_WINDOW command selects the non-current window as the current window.

You must execute the TWO_WINDOWS command before you can use the OTHER_WINDOW command to move from one window to the other. If you execute the OTHER_WINDOW command when there is only one window on the screen, the Monitor Interface displays a message stating that you have only one window.

If the mouse device is enabled, move the mouse pointer to the other window and click MB1 to change the current window.

Example

```
Command:  ONE_WINDOW [Return]
Command:  OTHER_WINDOW [Return]
```

The TWO_WINDOWS command divides the current window into two windows of equal size on your screen. The OTHER_WINDOW command changes the current window. VCS indicates the change by moving the asterisk (*) in the log window banner.

OUTPUT

Sends either a text string or a break signal to one or more serviced nodes, while retaining the Monitor Interface display.

Format

OUTPUT { *(/qualifier) node-name(, . . .) (string)* } { **/BREAK** *node-name(, . . .)* }

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

string
is the text string of up to 120 characters in length to be sent. The text string is not allowed if you specify the /BREAK qualifier. Quotation marks ("") are not required as string delimiters, although you may include them. For instance, the following two commands are equivalent and send the string **SET TIME=13:30** to ALPHA.

```
Command: OUTPUT ALPHA SET TIME=13:30
Command: OUTPUT ALPHA "SET TIME=13:30"
```

Quotation marks must be included if you want to do any of the following:

- Send lowercase characters. For example, the following command sends the string **Abcde** to node ALPHA:

  ```
  Command: OUTPUT ALPHA "Abcde"
  ```

- Send an exclamation point (!). For example, the following command causes the string **AB!CDE** to be sent to node ALPHA:

  ```
  Command: OUTPUT ALPHA AB"!"CDE
  ```

  Note that an exclamation point, if not enclosed in quotation marks, is considered to be a comment character, and any text following the exclamation point is ignored.

- Send a quotation mark ("). For example, the following command sends the string **"quoted"** to node ALPHA:

  ```
  Command: OUTPUT ALPHA ""quoted""
  ```

You can include special characters (such as Ctrl/Y or Ctrl/C) in the output string by prefixing them with Ctrl/V. For example, the following commands send a Ctrl/C to node HSC001:

```
Command: OUTPUT/NOTERMINATE HSC001 Ctrl/V Ctrl/C
Command: OUTPUT HSC001 SHOW DISKS
```

The first command sends a Ctrl/C to the HSC50 called HSC001, preparing it for the SHOW DISKS command that follows.

Qualifiers

/BREAK
indicates a break signal to be generated on the console line of the specified nodes. The /BREAK qualifier is not compatible with other qualifiers.

/TERMINATE (D)
/NOTERMINATE
indicates if a carriage-return character is to be included as part of the text string you send to the specified node. Usually you would include a carriage-return character to complete the text string as a command.

/NOTERMINATE indicates that the carriage-return character is not to be included as part of the text string you send to the specified node. Use this qualifier when you send a control character.

This qualifier is not compatible with the /BREAK qualifier.

The default is /TERMINATE.

OUTPUT

Description

Use the OUTPUT command to send either a text string or a break signal to one or more serviced nodes, while retaining the Monitor Interface display and continuing to view other console logs and monitor events.

When you send data to a serviced node such as a VMS system, make sure that the console terminal on that node is logged in, so that the node can process the text strings that you send.

You may want to send more than one command to a set of serviced nodes, or you may want to send a break signal and a series of commands to more than one serviced node. In this case, you can create a command file containing the OUTPUT commands that you wish to issue. Use the @ command to execute the command file. You may want to insert the WAIT command between the appropriate OUTPUT commands to allow the serviced system time to process the text strings.

If you are going to send more than one command to just one serviced node, you may want to use the SELECT or CONNECT commands instead of the OUTPUT command.

Examples

1 `Command: OUTPUT ALPHA, BETA, THETA SET TIME=6:20` Return

The OUTPUT command above simultaneously sends the text string **SET TIME=6:20** to ALPHA, BETA, and THETA, causing each node to set the time to 6:20 a.m. The Monitor Interface display is retained so that logs can continue to be viewed and events can continue to be monitored while the SET TIME command is executed on the nodes.

2 `Command: OUTPUT/BREAK ALPHA, BETA, THETA` Return

This OUTPUT command simultaneously sends a break signal to ALPHA, BETA, and THETA while retaining the Monitor Interface display.

PREVIOUS EVENT

Navigates backward in time the list of events in the Event Log or Event Context window.

Format

PREVIOUS EVENT

Description

The PREVIOUS EVENT command is valid only if the current window is either the Event Log window or the Event Context window. If you enter the PREVIOUS EVENT command in any other window, an error message is issued. Event order is determined by the order of the events in the Event Log buffer. The previous older event is the next event toward the top of the Event Log buffer, from the current event.

If the current window is the Event Log window, the PREVIOUS EVENT command makes the next older event the current event. The current event is highlighted according to its priority, as described in Section 6.1.2.1.1. If the current window is the Event Context window, the PREVIOUS EVENT command displays the context of the next older event, while at the same time making that event the current event.

The Monitor Interface maintains approximately 300 of the latest events. When this limit has been reached, as new events are detected and added to the buffer, the oldest of the currently maintained events are discarded. If you enter the PREVIOUS EVENT command and no older event exists, a warning is issued and the current event does not change. You can use the REVIEW/EVENT command to retrieve the older events from the event log files.

Examples

1
```
Command:  VIEW/EVENT  Return
Command:  PREVIOUS EVENT  Return
```

This sequence of commands enables you to view the Event Log window, then makes the next older event the current event. As you issue each of these commands, notice that the current event is changed. The highlighting of the event text shows you which event is current.

2
```
Command:  VIEW/CONTEXT  Return
Command:  PREVIOUS EVENT  Return
```

This sequence of commands enables you to view the Event Context window and then view the context of the next older event before the current event. If no older event exists in your Monitor Interface session, a warning is generated.

3
```
Command:  VIEW/EVENT  Return
Command:  TWO_WINDOWS  Return
Command:  VIEW/CONTEXT  Return
Command:  PREVIOUS EVENT  Return
```

This sequence of commands enables you to view the Event Log window, then split the screen into two windows and view the Event Context window in the current window of the two windows. Issuing the PREVIOUS EVENT command causes the next older event to become the current event. You will see it become highlighted in the Event Log window. The same PREVIOUS EVENT command causes the Event Context window (the current window) to display the context of the event which has just become the current event.

PREVIOUS SCREEN

Moves the current window backward in time by one window over the current buffer information so that the previous window of data is visible.

Format

PREVIOUS SCREEN

Description

The PREVIOUS SCREEN command moves the log window from the current position in the buffer to the previous window of information. If there is less than one window of previous data available, this command has the same result as if you issued the GOTO FIRST command.

You can also execute the PREVIOUS SCREEN command by pressing Prev Screen on the VT200-series editing keypad or KP. on the VT100 numeric keypad.

If you specify PREVIOUS, the command defaults to PREVIOUS SCREEN.

Example

```
Command:  REVIEW AUTO [Return]
Command:  PREVIOUS SCREEN [Return]
```

This command displays console log information for the node named AUTO. When you type PREVIOUS SCREEN, the Monitor Interface moves the current window backward one window toward the oldest log information.

RECONFIGURE

Notifies VCS of changes that the system manager made in the VCS configuration file, and requests VCS to attempt to repair the host initiated LAT connections that were interrupted.

Format

RECONFIGURE

Description

When VCS starts up, it reads in the configuration file contents and retains that information for managing the console lines and the various user interfaces. That information remains static to the VCS processes. You may, at any time, update the configuration file, using the Configuration Editor, without affecting VCS's current knowledge about its environment and the entities it services.

To notify VCS of the changes that you made to the configuration file, you must issue the RECONFIGURE command after successfully updating the information. Otherwise, that information remains ineffective until VCS is shut down and restarted.

You must issue the SET PRIVILEGES command before you can enter the RECONFIGURE command. When you issue the RECONFIGURE command, if the configuration file has not been changed since the last time VCS processed its contents, nothing changes. The RECONFIGURE command affects the VCS operations as follows:

- If you changed the scan profile definition in the configuration file, the new definition is in effect when the reconfiguration completes successfully.
- If you changed a node name, the old node is no longer in the VCS configuration. The logical name VCS$ALL_NODES is redefined to reflect the new list of node names.

- If you changed the terminal device associated with a node, that node is considered logically unchanged (for example, data from the new terminal line is appended to the previously recorded data from the old terminal line).
- A disabled system retains its disabled status.
- VCS notifies any running interfaces that a reconfiguration has occurred. If a node has been added to the configuration, an interface must be started again to gain access to the new node.
- If placement of the icons in the Central Control Coordinator Interface has been changed, the new placement is reflected in the currently active displays. New configuration information is available to the C³ Show Information windows.
- Changes to the ENS data file become active.

RECONFIGURE does not affect the definition of the logical name VCS$LOG. The new definition will not take effect until you shut down and restart the VCS software.

The RECONFIGURE command also requests VCS to attempt to repair any LAT host initiated connections that were interrupted.

Example

```
Command:  SET PRIVILEGES [Return]
Command:  RECONFIGURE [Return]
```

The first command enables the privileges. The second command notifies VCS of changes made to the VCS configuration file.

REFRESH

Eliminates extraneous characters from your screen and restores the screen to the previous display.

Format

REFRESH

Description

In addition to clearing extraneous characters from the screen, this command also erases any messages on the message line. You can issue the REFRESH command by pressing Ctrl/W.

REVIEW

Activates the Review Interface to allow you to examine the console and event data contained in the VCS console and event log files, for a specific set of serviced nodes over a specific period of time.

Format

REVIEW { *node-name(, . . .) (/qualifier)* | **/PREVIOUS** }

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

The default node name is the one that you are viewing in the *current* window.

Qualifiers

/PREVIOUS
indicates that you want to examine the same information that the previous REVIEW command already retrieved from the log files. This qualifier is incompatible with all others.

/BEFORE=date-time
is the ending point in the time period for which data is to be retrieved. The time and date must follow the VMS convention and in the format:

```
[dd-mmm-yyyy[:]][hh:mm:ss.cc]
```

This time must not predate the available log files and must not be in the future. It must always be a later time than the time specified by the /SINCE qualifier. If /BEFORE is not specified, then the ending point of the time period defaults to the current date and time. If you specify a time without the date, it defaults to today at the specified time. If you specify a date without the time, it defaults to the specified date at 00:00:00.00.

/EVENT= [{ ***CONTEXT*** }]
[{ ***NOCONTEXT (D)*** }]
[{ ***EVENT_NAME (D)*** }]
[{ ***NOEVENT_NAME*** }]

indicates if event data is to be reviewed. If /EVENT is not specified, console data is reviewed

CONTEXT specifies that the context lines for each event be included in the output. Specify NOCONTEXT if you do not want the context lines included. The default is /EVENT=NOCONTEXT.

EVENT_NAME specifies that the event name, as specified by the system manager in the VCS scan files, should be included in the event line in the output. The default is /EVENT=EVENT_NAME.

/SINCE=date-time
is the beginning point in the time period for which data is to be retrieved. The time and date must follow the VMS convention and in the format:

```
[dd-mmm-yyyy[:]][hh:mm:ss.cc]
```

This time must not predate the available log files and must not be in the future. It must always be an earlier time than the time specified by the /BEFORE qualifier. If /SINCE is not specified, then the starting point of the time period defaults to the current date and 00:00:00.00. If you specify a time without the date, it defaults to today at the specified time. If you specify a date without the time, it defaults to the specified date at 00:00:00.00.

/SYSTEM_NAME (D)
/NOSYSTEM_NAME
indicates whether the name of the serviced node associated with each line of data is to be included in the output. /NOSYSTEM_NAME is useful when reviewing data for a single node only.

The default is /SYSTEM_NAME.

/TIME_STAMP (D)
/NOTIME_STAMP
indicates whether the time stamp associated with each line of data is to be included in the output.

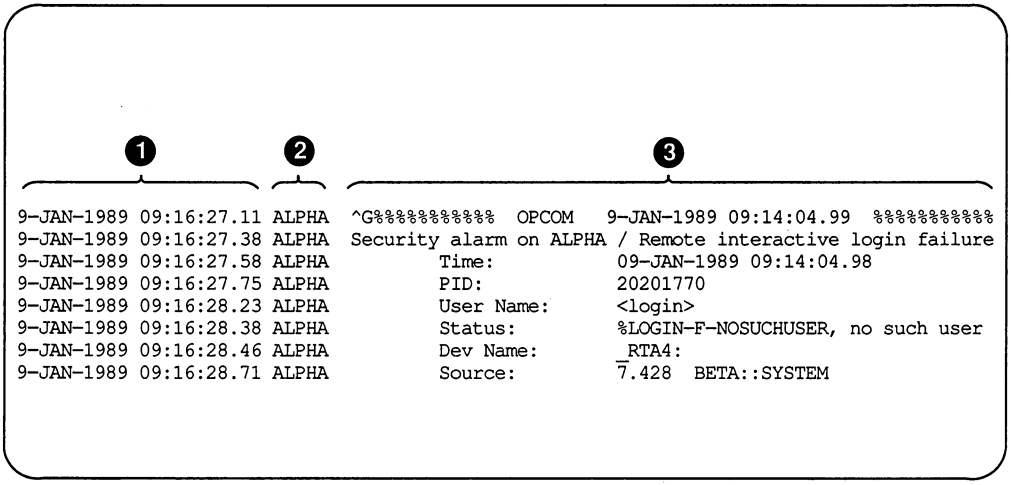
The default is /TIME_STAMP.

Description

Use the REVIEW command to retrieve the log data in the VCS console or event log files for a specific period of time. This command allows you to examine the log data that is no longer available in the log buffers of the Monitor Interface, as well as data that is current in the log buffers. The retrieved data is displayed in the review log buffer.

Retrieving console data is the default function. If you want to examine event data, you must use the /EVENT qualifier. Figure 6–6 contains the description of the console data as displayed by the Review Interface.

Figure 6–6 Reviewed Text Anatomy


❶
❷
❸
9-JAN-1989 09:16:27.11 ALPHA ^G%%%%%%%%%% OPCOM 9-JAN-1989 09:14:04.99 %%%%%%%%%%
9-JAN-1989 09:16:27.38 ALPHA Security alarm on ALPHA / Remote interactive login failure
9-JAN-1989 09:16:27.58 ALPHA Time: 09-JAN-1989 09:14:04.98
9-JAN-1989 09:16:27.75 ALPHA PID: 20201770
9-JAN-1989 09:16:28.23 ALPHA User Name: <login>
9-JAN-1989 09:16:28.38 ALPHA Status: %LOGIN-F-NOSUCHUSER, no such user
9-JAN-1989 09:16:28.46 ALPHA Dev Name: _RTA4:
9-JAN-1989 09:16:28.71 ALPHA Source: 7.428 BETA::SYSTEM

MR-4079-RA

The following descriptions are keyed to Figure 6–6.

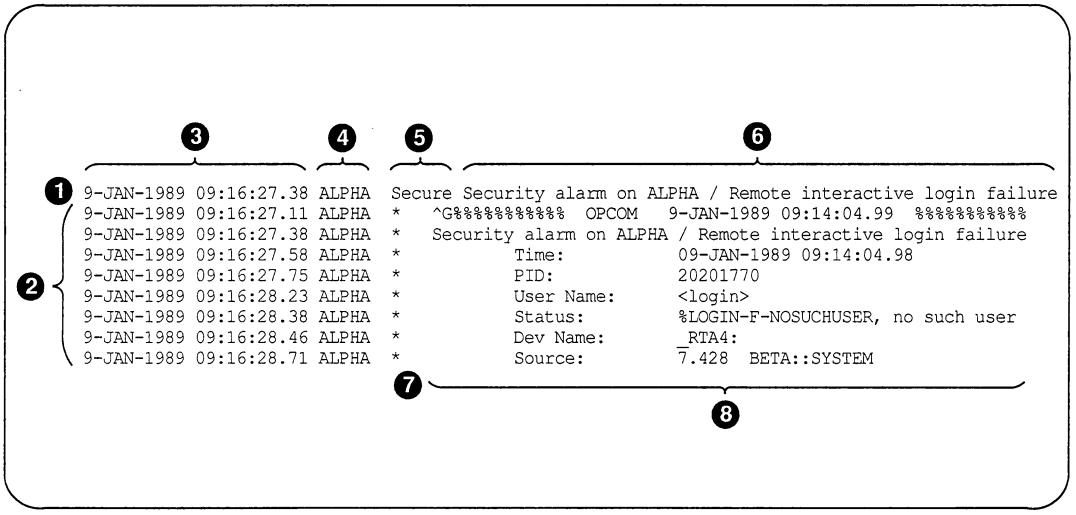
❶ The time stamp that is associated with the data line. If you want to omit the time stamp, specify /NOTIME_STAMP.

❷ The name of the serviced system from which the console data was received. If you want to omit the system name, specify /NOSYSTEM_NAME.

❸ The console data text as received from the serviced system.

The /EVENT qualifier allows you to retrieve the event data. Figure 6–7 contains the description of the event data as displayed by the Review Interface.

Figure 6–7 Reviewed Event Anatomy


❸
❹
❺
❻
❶
❷
9-JAN-1989 09:16:27.38 ALPHA Secure Security alarm on ALPHA / Remote interactive login failure
9-JAN-1989 09:16:27.11 ALPHA * ^G%%%%%%%%%% OPCOM 9-JAN-1989 09:14:04.99 %%%%%%%%%%
9-JAN-1989 09:16:27.38 ALPHA * Security alarm on ALPHA / Remote interactive login failure
9-JAN-1989 09:16:27.58 ALPHA * Time: 09-JAN-1989 09:14:04.98
9-JAN-1989 09:16:27.75 ALPHA * PID: 20201770
9-JAN-1989 09:16:28.23 ALPHA * User Name: <login>
9-JAN-1989 09:16:28.38 ALPHA * Status: %LOGIN-F-NOSUCHUSER, no such user
9-JAN-1989 09:16:28.46 ALPHA * Dev Name: _RTA4:
9-JAN-1989 09:16:28.71 ALPHA * Source: 7.428 BETA::SYSTEM
❼
❽

MR-4080-RA

The following descriptions are keyed to Figure 6–7.

❶ The event information that was recorded in the event log file.

❷ The event context information, if any are available. The event context is displayed only when /EVENT=CONTEXT is specified.

❸ The time stamp that is associated with the data line. If you want to omit the time stamp, specify /NOTIME_STAMP.

❹ The name of the serviced system where the console data was received. If you want to omit the system name, specify /NOSYSTEM_NAME.

❺ The name of the event, as specified by the system manager in the scan files. If you want to omit the event name, specify /EVENT=NOEVENT_NAME.

❻ The console data text as received from the serviced system. It contains the text string that matches a predefined one as specified by the user in a scan event record. This text line may show a partial console text line if the Data Scanner notifies the Event Logger of a match before the remainder of the console text line is received from the serviced node.

❼ The delimiter "*" indicates the event context data.

❽ The console data text as received from the serviced system.

Examples

1 `Command: REVIEW/NOTIME_STAMP ALPHA, BETA` Return

This command requests that the contents of the console log file for the current day be reviewed from the beginning of the day until the current time for the nodes ALPHA and BETA.

2 `Command: REVIEW/EVENT=CONTEXT/SINCE=14:20 VCS$ALL_NODES` Return

This command requests that the contents of the event log file for the current day be reviewed from 2:20 p.m. until the current time for all the nodes in the current VCS configuration (logical name VCS$ALL_NODES). Also, the context information for each event is included in the output.

SAVE

Saves the Monitor Interface display attributes and key definitions.

Format

SAVE *(file-name)*

Parameters

file-name
is the name of the file storing the display attributes and key definitions defined during the current session.

The default file name is:

default-device:[*default-directory*]VCS$MONITOR_SECTION.TPU$SECTION

Description

The SAVE command saves key definitions you have created with the DEFINE/KEY, SET SHIFT_KEY, and UNDEFINE/KEY commands. It also saves the break key and escape key defined with the SET BREAK and SET ESCAPE commands.

Other Monitor Interface display attributes such as bell tone indicator, mouse locator usage, time stamp display indicator, terminal screen width, and so on, are also saved by the SAVE command. See the SET and SHOW commands for more information about these attributes.

The key definitions and display attributes make up the Monitor Interface environment. This environment may be restored the next time you invoke the Monitor Interface by specifying the /SECTION qualifier with the VCS MONITOR command. If you use the same file name with each SAVE command and with the /SECTION qualifier to each VCS monitor command, your definitions will be cumulative from one VCS session to another.

If you want to make the saved environment available system wide, specify SYS$LIBRARY:VCS$MONITOR_SECTION.TPU$SECTION as the file name. Exit the Monitor Interface, and at DCL level, enter the following command to install the section image file:

```
$  @SYS$STARTUP:VCS$STARTUP INSTALL [Return]
```

Now, the users have the new environment without having to specify the /SECTION qualifier when they invoke the Monitor Interface.

Example

```
Command:  SAVE DOC1:[KATE]VCSKEYDEFS.TPU$SECTION [Return]
%VCS-S-SECTSAVED, 116 procedures,160 variables,33 keys saved
Command:  EXIT [Return]
$  VCS MONITOR/SECTION=DOC1:[KATE]VCSKEYDEFS.TPU$SECTION - [Return]
_$  ALPHA,BETA [Return]
```

This example shows how to store key definitions created during a current session in the file DOC1:[KATE]VCSKEYDEFS.TPU$SECTION. If you want to save your key definitions, you must issue this command before exiting from the Monitor Interface. When you save the key definitions, the Monitor Interface displays a message which verifies that the definitions are saved. To access the file containing the key definitions in a subsequent VCS session, use the VCS MONITOR/SECTION command.

SCROLL

Moves the current window up or down vertically (backward or forward in time, respectively) by the specified number of lines.

Format

SCROLL { **DOWN** / **UP** } *count*

Parameters

DOWN
indicates the current window is to move down (or forward in time), by the specified number of lines, over the current buffer information so that the next number of lines of data is visible. The SCROLL DOWN command is bound to the KP2 key.

UP
indicates the current window is to move up (or backward in time), by the specified number of lines, over the current buffer information so that the previous number of lines of data is visible. The SCROLL UP command is bound to the KP5 key.

count
is the number of lines to scroll. The default is one.

Description

If the mouse device is enabled, move the mouse pointer to the window and click and drag MB1 in the direction you want the window to scroll.

SCROLL

Example

```
Command:  SCROLL UP 5 Return
```

This command moves the current window up five lines through the currently selected log.

SELECT

Connects your keyboard to a node and simultaneously retains the Monitor Interface display on the screen.

Format

SELECT { *node-name* } { */qualifier* }

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

The default node name is the one that you are viewing in the *current* window.

Qualifiers

/ACTIVE
indicates the first node, leftmost one, listed in the event summary window as the node you wish to communicate with.

/NEXT
indicates the subsequent node in the list, which you specified with the VCS MONITOR command, as the node you want to communicate with.

SELECT

Description

Use the SELECT command to communicate with a node while you continue to monitor the events for the other serviced nodes.

When you enter the SELECT command, the cursor moves to the log window of the node. The escape and break character definitions are displayed in the command window for the duration of the SELECT command. Also, while the SELECT command is in effect, the Monitor Interface does not display event indicators for the selected node in the event summary window. However, those events are still being recorded and displayed in the Event Log buffer and can be viewed later by using the VIEW/EVENT command.

Now, the Monitor Interface sends all characters typed (excluding the break character, escape character, Ctrl/S, and Ctrl/Q) to the current system console line. While the SELECT command is in effect, the key definitions of the node you have selected take precedence over your Monitor Interface key definitions. All of the key definitions for the Monitor Interface are ignored while you are in this mode. In addition, the Monitor Interface maintains control of the display on your terminal screen. If you attempt to run a screen-oriented text editor or program that requires control of the screen, screen-control characters output by the node are displayed in the log window as control characters. Thus the screen may look garbled.

When you finish typing commands to the node, you return to the command line by pressing the escape character. The default escape character is Ctrl/G. You can change the escape character with the SET ESCAPE command.

Unlike the Connect Interface, while you are selecting a node, typing the break character will immediately generate a Break signal on the console line. You will not be prompted to confirm whether you intend to do so. You can set and change the break character with the SET BREAK command.

Example

```
Command:  SELECT HSC001 Return
```

This command enables you to type commands to node HSC001 while you monitor events for the other nodes in the VCS configuration.

SET BELL

Enables or disables the bell tones associated with the arrival of an event indicator in the event summary window.

Format

SET (NO)BELL

Description

The SET [NO]BELL command enables and disables the bell tone which is sounded when an event indicator is posted in the event summary window. By default, the bell tone is set on, so the bell rings when event indicators arrive.

Use the SHOW MONITOR command to show the value of the bell tone attribute. The SAVE command saves the setting of the bell tone attribute.

Example

```
Command:  SET BELL Return
```

This command enables the bell tone so that when an event indicator is displayed in the event summary window, the bell tone is sounded.

SET BREAK

Defines a control key that represents the Break key. When it is pressed, a Break signal is generated on the console line to the serviced node.

Format

SET { **BREAK** *key-name* | **NOBREAK** }

Parameters

key-name
is the name of the key or control character that represents the Break key. Use the key names specified in Table 6–3 and Table 6–5 to define the break character. You cannot define the Return key and all printable alphanumeric characters. Also, it is recommended that you do not redefine the control characters specified in Table 6–4. If you do redefine one of these control characters, you cannot access the function originally associated with it during the current Monitor Interface session.

Description

There is no default break key. This break key is not the same as the Break key on your keyboard. Pressing the key labeled Break on your keyboard does not generate a Break signal on the console line to the serviced node, but is interpreted by your VCS host system or the communications interface device that your terminal is connected to and may interrupt your session.

To generate a Break signal on the console line to the serviced node, you must use the break key you define with SET BREAK. For example, you type the SELECT command to connect to a node, then type commands to that node. You must use the break key you define with SET BREAK to generate a Break signal on the console line to that node.

SET BREAK

See the descriptions of the SELECT and CONNECT commands for how each handles the break key when it is in effect.

You can use the SHOW MONITOR command to show the setting of the Break character. The SAVE command saves the setting of the Break character.

Example

```
Command:  SET BREAK CTRL_L_KEY [Return]
Command:  SELECT ALPHA [Return]
```

This command enables you to set the break key to be Ctrl/L. You enter SELECT ALPHA to connect to the node named ALPHA, then press Ctrl/L to generate a Break signal on the console line to ALPHA.

SET ESCAPE

Defines a control key that is used to disconnect from a node.

Format

SET ESCAPE *key-name*

Parameters

key-name
is the name of the key or control character that is to be used as the escape key. Use the key names specified in Table 6–3 and Table 6–5 to define the escape character. You cannot define the Return key and all printable alphanumeric characters. Also, it is recommended that you do not redefine the control characters specified in Table 6–4. If you do redefine one of these control characters, you cannot access the function originally associated with it during the current Monitor Interface session.

Description

The SET ESCAPE command defines a key as the escape character used to disconnect from a node.

For example, enter the SELECT command to connect to a node and type commands to that node. When you finish typing commands to that node, press the escape character that you defined with the SET ESCAPE command to return to the Monitor Interface command line.

If you do not use the SET ESCAPE command to define an escape key, you can use the default escape character, Ctrl/G, to disconnect from a node.

You can use the SHOW MONITOR command to show the setting of the Escape character. The SAVE command saves the setting of the Escape character.

Example

```
Command:  SET ESCAPE CTRL_K_KEY Return
Command:  SELECT SATURN Return
```

This sets the escape key to be Ctrl/K. Enter SELECT SATURN to connect to the node named SATURN. Now, enter commands to SATURN. When you are finished entering commands to SATURN, press Ctrl/K to disconnect from SATURN and return to the Monitor Interface command line.

SET LOCK

Defines the various modes of access locking for communications with the serviced nodes.

Format

SET LOCK { (NO)AUTO | OFF | ON }

Parameters

AUTO
indicates, from this moment on, that the nodes are automatically locked if you enter the SELECT or CONNECT command to communicate with a node. This is analogous to entering the SET LOCK ON command before issuing the SELECT or CONNECT command, and the SET LOCK OFF command afterward.

Note that automatic locking is not set for the OUTPUT command.

OFF
indicates that the lock on the node in the current log window is released and that you now share input rights with other users to the specified node. From this moment on, any other operator can enter data simultaneously to the system.

ON
indicates that you intend to reserve the exclusive input rights to the node in the current log window. If you do so, other users cannot send data to the node, but they can still examine log information and view your current activity. The node remains locked until you explicitly release it with the SET LOCK OFF command or when you terminate the current Monitor Interface session.

SET LOCK

Description

If you set a lock on a node, you reserve the exclusive input rights to that node. If you do so, other users cannot send data to the node, but they can still examine log information and view your current activity. When you release a lock on a node, you share input rights with other users to that node. Any other operator would be able to enter data simultaneously to that node.

If you enter the SET LOCK AUTO command, a node remains locked until you disconnect from the node or select another node. When you exit the Monitor Interface, all the locks which you have set are released.

If you lock a node to which another user is connected, VCS sets the lock and issues a message to the other user when he or she tries to send a command to the node. If you try to lock a node that another user has locked, VCS displays an error message.

Use SHOW MONITOR to show the status of the automatic lock attribute. Use the SHOW NODES to determine the lock status of the nodes you are monitoring. You cannot save the lock setting with the SAVE command.

Example

```
Command:  SELECT SATURN [Return]
Command:  VIEW PLUTO [Return]
Command:  SET LOCK ON [Return]
Command:  OTHER_WINDOW [Return]
Command:  SHOW NODES [Return]
```

First, issue the TWO_WINDOW command to establish two log windows on the screen. When you use the command VIEW PLUTO to view console log information, the node PLUTO becomes the current node. SET LOCK ON sets a lock on the current node, PLUTO, so other users cannot send commands to that node. When you switch to the other window and request the node list, VCS displays a table that includes the node's lock status.

SET MOUSE

Enables or disables the VAXstation mouse capability as a locator device.

Format

SET (NO)MOUSE

Description

SET MOUSE enables the VAXstation mouse to be used as a locator device for the Monitor Interface. The SET MOUSE command disables the DECwindows terminal emulator's cut and paste feature of the DECterm window in which the Monitor Interface is running. SET NOMOUSE disables the mouse locator.

The default setting of the Mouse Locator is ON.

Use the SHOW MONITOR command to show the status of the mouse locator attribute. The SAVE command saves the setting of the Mouse Locator attribute.

Example

```
Command:  SET MOUSE Return
```

The SET MOUSE command enables the mouse device to be used as a locator device. You can use the mouse device to scroll the window up and down, shift the window left and right, and select the current window.

SET PRIVILEGES

Makes the privileged VCS system control functions available to you in the Monitor Interface. These privileged commands are ENABLE, DISABLE, RECONFIGURE, and SHUTDOWN.

Format

SET (NO)PRIVILEGES

Description

The SET PRIVILEGES command enables you to use the VCS system control commands at the Monitor Interface level. The original setting for each Monitor Interface session is always SET NOPRIVILEGES. You cannot save this attribute with the SAVE command.

If the system manager has not set up your account with VCS privileges, then issuing the SET PRIVILEGES command has no effect.

Even though VCS access may not be restricted on your system, you must still enter the SET PRIVILEGES command before you can execute the VCS system control commands.

Example

```
Command:  SET PRIVILEGES Return
```

The VCS privileges are enabled.

SET SHIFT_KEY

Defines a control key that is to be used as the *shift* key.

Format

SET { **SHIFT_KEY** *key-name* | **NOSHIFT_KEY** }

Parameters

key-name
is the name of the key or control character that is to be used as the shift key. Use the key names specified in Table 6–3 and Table 6–5 to define the shift key. You cannot define the RETURN key and all printable alphanumeric characters. Also, it is recommended that you do not redefine the control characters specified in Table 6–4. If you do redefine one of these control characters, you cannot access the function originally associated with it during the current Monitor Interface session.

Description

The SET SHIFT_KEY command enables you to define a key as a *shift* key. This shift key is not the same as the Shift key on your keyboard. The shift key acts as the prefix key which alters the meaning of the key that follows it.

This command is related to the DEFINE/KEY command. If you have a shift key defined, you can assign two commands to the same key with the DEFINE/KEY command. One command is executed when the defined key is preceded by the shift key and the other when the defined key is pressed individually.

Use the SHOW MONITOR command to show the shift key definition. The SAVE command saves the setting of the shift key.

Example

```
Command:  SET SHIFT_KEY KP8 Return
Command:  DEFINE/KEY/TERMINATE/SHIFT_KEY KP9 SHOW DIRECTORY Return
```

This command sequence first sets the key KP8 to be the shift key. Then the DEFINE/KEY command binds the SHOW DIRECTORY command to the shifted KP9. When you press the shift key KP8 followed by KP9, you execute the SHOW DIRECTORY command.

SET TIME_STAMP

Determines if the time stamp is displayed at the beginning of each line in a log window.

Format

SET (NO)TIME_STAMP

Description

VCS precedes each line in a log buffer with a time stamp. The SET TIME_STAMP command causes the time stamp to be included in the Monitor Interface display. If you issue this command, you may want to increase the width of your terminal display with the SET WIDTH command.

The SET NOTIME_STAMP command eliminates from the display the time stamps that are included at the beginning of each line in a log buffer. However, the time stamp continues to be written to the log file. By default, the time stamps are not displayed.

Use the SHOW MONITOR command to show the setting of the time stamp attribute. The SAVE command saves the setting of the time stamp attribute.

Example

```
Command:  SET TIME_STAMP  Return
Command:  VIEW PLUTO  Return
```

By default, VCS does not display time stamps when you display log information with the VIEW command. Enter SET TIME_STAMP to display the time stamps. Then enter VIEW PLUTO.

```
10:00:29.39 %%%%%%%%%%  OPCOM   4-APR-1986 09:59:57.22  %%%%%%%%%%
10:00:29.78 Message from user INTERNETWORK on Pluto
10:00:30.21 Configure done
```

The previous example shows a representative display with time stamps from the log of node PLUTO.

SET WIDTH

Adjusts the width, in characters, of the terminal display.

Format

SET WIDTH *columns*

Parameters

columns
is the number of columns to be displayed on the terminal screen. The number of columns ranges from 1 to 132. The widths 80 and 132 are most useful for VCS.

Description

Set the width of your terminal to 80 or 132 with the SET WIDTH command. It may be necessary to widen your terminal width to display log information wider than 80 characters, such as, log information displayed with time stamps.

The SET WIDTH command affects all Monitor Interface windows and all subsequent windows, not just the one you are currently viewing. For example, if you are viewing a log and set the width to 132, and then view the HELP buffer, the HELP buffer display is 132 characters wide.

Example

```
Command:  SET WIDTH 132 Return
```

This command sets the width of your terminal display to 132 characters.

SHIFT

Moves the current window to the left or right over the current buffer information so that the data outside of the current margin is visible.

Format

SHIFT { **LEFT** / **RIGHT** } *columns*

Parameters

LEFT
indicates the log window is to move to the left. The SHIFT LEFT command is bound to the KP1 key.

RIGHT
indicates the log window is to move to the right. The SHIFT RIGHT command is bound to the KP3 key.

columns
is the number of columns to move the log window.

The default number of columns is 10.

Description

The SHIFT command moves the log window(s) to the left or right so that you can view log information that is too wide to fit on one screen.

When you use the SHIFT command to move the log window, all buffers subsequently displayed are also shifted. If you are currently in two-window mode, both log windows are also shifted in unison.

If the mouse device is enabled, click and drag MB2 in the direction you want the log window to be shifted.

Examples

1 `Command: SHIFT RIGHT 40` Return

This command causes the current window to be shifted 40 spaces to the right.

2 `Command: SHIFT LEFT 20` Return

This command causes the current window to be shifted 20 spaces to the left.

SHOW DIRECTORY

Displays information about the log files available on disk.

Format

SHOW DIRECTORY *(/qualifier)*

Qualifiers

/ALL (D)
Displays the names and dates of both console and event logs. The /ALL qualifier cannot be specified with any other qualifier.

This is the default.

/CONSOLE
Displays the names and dates of console logs only. The /CONSOLE qualifier can be combined with the /EVENT qualifier.

/EVENT
Displays the names and dates of event logs only. The /EVENT qualifier can be combined with the /CONSOLE qualifier.

Description

The SHOW DIRECTORY command displays information about the log files available in the current VCS session. The information displayed is:

- The log directory
- Log file date
- Log file name

Examples

1 Command: SHOW DIRECTORY/ALL Return

This command displays the names and dates of both console and event logs, as follows:

```
Log files based on logical name VCS$LOG
   contained in directory SYS$SYSROOT:[SYSMGR]
     Date        |          Filename
 ----------------------------------------
  11-JUL-1988    |    VCS$1988_07_11.LOG;1
  12-JUL-1988    |    VCS$1988_07_12.LOG;1
  13-JUL-1988    |    VCS$1988_07_13.LOG;1

Event log files based on logical name VCS$LOG
   contained in directory SYS$SYSROOT:[SYSMGR]
     Date        |          Filename
 ----------------------------------------
  11-JUL-1988    |    VCS$EVENT_1988_07_11.LOG;1
  12-JUL-1988    |    VCS$EVENT_1988_07_12.LOG;1
  13-JUL-1988    |    VCS$EVENT_1988_07_13.LOG;1
```

2 Command: SHOW DIRECTORY/EVENT Return

This command displays the names and dates of event logs only, as follows:

```
Event log files based on logical name VCS$LOG
   contained in directory SYS$SYSROOT:[SYSMGR]
     Date        |          Filename
 ----------------------------------------
  11-JUL-1988    |    VCS$EVENT_1988_07_11.LOG;1
  12-JUL-1988    |    VCS$EVENT_1988_07_12.LOG;1
  13-JUL-1988    |    VCS$EVENT_1988_07_13.LOG;1
```

3 Command: SHOW DIRECTORY/CONSOLE Return

This command displays the names and dates of console logs only, as follows:

```
Log files based on logical name VCS$LOG
   contained in directory SYS$SYSROOT:[SYSMGR]
     Date        |          Filename
 ----------------------------------------
  11-JUL-1988    |    VCS$1988_07_11.LOG;1
  12-JUL-1988    |    VCS$1988_07_12.LOG;1
  13-JUL-1988    |    VCS$1988_07_13.LOG;1
```

SHOW KEYS

Displays the current key definitions.

Format

SHOW KEYS *(/qualifiers)*

Qualifiers

/COMMAND
indicates that the Monitor Interface commands that defined keys are to be displayed. This qualifier lists only the locally defined keys. You can save these commands to a text file with the WRITE command.

/DEFINITION (D)
The /DEFINITION qualifier displays the key name, key modifier, and functional description of all defined keys.

If the key was defined with the /SHIFT_KEY qualifier, the shift key precedes the key name either as <Shift>, or, if a shift key has been set, the actual key name is in angle brackets. For example, <PF1>.

The key modifier is either **Digital** or the name of the user who last modified or created the key definition.

If the key was originally defined by Digital, the functional description briefly describes the function. If you define the key, the functional description is the text string bound to the key. If the text string is bound with the /TERMINATE qualifier, it is punctuated with a blank space followed by a Return.

Description

When you enter the SHOW KEYS command, VCS displays information about the current key definitions on your screen in the current window.

SHOW KEYS

Examples

1 Command: SHOW KEYS/DEFINITION Return

VCS displays the following:

```
MODIFIER     DEFINED KEY NAMES     DESCRIPTION / FUNCTION

Digital      CTRL/A ............. Change insert mode
Digital      CTRL/B ............. Recall previous command
Digital      CTRL/D ............. Move cursor left
Digital      CTRL/E ............. Move cursor to end of line
Digital      CTRL/F ............. Move cursor right
Digital      CTRL/H ............. Move cursor to begin of line
Digital      CTRL/I ............. Tab
Digital      CTRL/J ............. Erase previous word
Digital      RETURN ............. Execute
Digital      CTRL/T ............. Show current time
Digital      CTRL/U ............. Erase to the begin of line
Digital      CTRL/V ............. Quote next character
Digital      CTRL/W ............. Refresh the screen
Digital      CTRL/X ............. Erase to the begin of line
Digital      CTRL/Z ............. Exit the Monitor Interface
Digital      DELETE ............. Delete
Digital      UP ................. Recall previous command
Digital      DOWN ............... Recall next command
Digital      RIGHT .............. Move cursor right
Digital      LEFT ............... Move cursor left
Digital      PF1 ................ Find
Digital      PF2 ................ Help
Digital      PERIOD ............. Previous screen
Digital      KP0 ................ Next screen
Digital      KP1 ................ Shift screen left 10 characters
Digital      KP2 ................ Next line
Digital      KP3 ................ Shift screen right 10 characters
Digital      KP5 ................ Previous line
OPERATOR     <PF1> KP5........... SHOW NODES
Digital      FIND ............... Find
Digital      PREV SCREEN ........ Previous screen
Digital      NEXT SCREEN ........ Next screen
Digital      F10 ................ Exit the Monitor Interface
Digital      F12 ................ Move cursor to begin of line
Digital      F13 ................ Erase previous word
Digital      F14 ................ Change insert mode
Digital      HELP ............... Help
Digital      DO ................. Enable/Disable Mouse Locator
```

2 Command: SHOW KEYS/COMMAND Return

VCS displays the following:

```
SET SHIFT PF1
DEFINE/KEY/SHIFT KP5 SHOW NODES
```

SHOW MONITOR

Displays information about the Monitor Interface and display attributes.

Format

SHOW MONITOR

Description

When you enter the SHOW MONITOR command, VCS displays information about the current Monitor Interface session on your screen in the current window.

If your privileges are not enabled (see SET PRIVILEGES command), then the privileges attribute is not included in the display.

SHOW MONITOR

Example

```
Command:  SHOW MONITOR Return
```

VCS displays the following:

```
Digital Equipment Corporation VAXcluster Console System

 Version                V1.3-000
 Creation Date          11-JAN-1990 20:34:29

 Section File           SYS$COMMON:[SYSLIB]VCS$MONITOR_SECTION.TPU$SECTION;16
  Created By            Digital Equipment Corporation
  Creation Date         Thursday January 11, 1990 at 20:35
  VAXTPU                Version 2.4

 Attributes
  Auto Lock             Off
  Bell Tone             On
  Mouse Locator         On
  Privileges            Enabled
  Time Stamp            Off

 Special Characters
  Break Character       No break key is set
  Escape Character      CTRL/G
  Shift Character       No shift key is set

 Display Information
  Screen Width          80 characters
  Shift Window          0 character positions to the right
```

SHOW NODES

Displays the information and status about the serviced nodes currently being monitored.

Format

SHOW NODES

Description

Use the SHOW NODES command to display information about and status of the serviced nodes currently being monitored by the Monitor Interface session. The information includes the node name, the interface device name, the scan profile, the lock status and whether service to the node is enabled or disabled.

The Monitor Interface displays information in the following format:

❶	❷	❸	❹	❺	❻	❼
NO	NODE NAME	DEVICE NAME	ENABLE STATUS	LOG STATUS	LOCK STATUS	SCAN PROFILE
1	AQUABL	LTA917	Yes	Yes	OPERATOR	VMS9000
2	FAMB05	LTA511	No	Yes	On	VMS6400
3	FAMC06	LTA521	Yes	Yes	Off	VMS6400
4	GALLON	LTA518	Yes	Yes	Off	VMS6400
5	HSC0	LTA515	Yes	No	FIELD	HSC70
6	HSC1	LTA516	Yes	No	Unknown	HSC70

The following descriptions are keyed to the SHOW NODES display:

❶ The order of the node names is the same as specified in the VCS MONITOR command and is the order used by the VIEW/NEXT command.

❷ The name that the system manager assigns to a node in the VCS configuration file.

❸ The name of the terminal device port that connects the VCS host system to the serviced node's console port.

❹ The current service status which may be modified dynamically by the DISABLE and ENABLE commands:

Value	Description
No	Service to the node has been disabled.
Yes	Service to the node is normal.

❺ The current log status which may only be modified by the Configuration Editor and activated at either startup or reconfiguration of VCS:

Value	Description
No	Data received from the node is not being recorded in the console log file.
Yes	Data received from the node is being recorded in the console log file.

❻ The current lock status which indicates the current exclusive output rights to a console port:

Value	Description
user-name	The node is currently being locked by the user *user-name*.
Off	Access to the node is not restricted.
On	The node is currently being locked by your Monitor Interface session.
Unknown	The process that owned the lock was aborted prior to its being able to release the lock on the node. The start of any new VCS user interface will clean up any old locks.

❼ The scan profile that is associated with the node.

Example

Command: SHOW NODES Return

This command displays the following information about the VCS nodes, including the device designator of the console line, the scan profile, the lock status, and whether the nodes are enabled or disabled:

NO	NODE NAME	DEVICE NAME	ENABLE STATUS	LOG STATUS	LOCK STATUS	SCAN PROFILE
1	AQUABL	LTA917	No	Yes	Off	VMS9000
2	FAMB05	LTA511	No	Yes	Off	VMS6400
3	FAMC06	LTA521	No	Yes	Off	VMS6400
4	GALLON	LTA518	Yes	No	SYSTEM	VMS6400
5	HSC0	LTA515	No	Yes	Off	HSC70
6	HSC1	LTA516	Yes	Yes	Off	HSC70
7	HSC2	LTA517	Yes	Yes	Off	HSC70
8	LETOUR	LTA513	No	Yes	OPERATOR	VMS6400
9	MUSSEL	LTA514	Yes	Yes	Off	VMSII
10	PROTO	LTA918	Yes	Yes	Off	VMS9000
11	VECTOR	LTA915	Yes	Yes	Off	VMS6400
12	YQUEMO	LTA512	Yes	No	Off	VMS6400

SHOW TIME

Displays the current time of day in the message window.

Format

SHOW TIME

Description

This command causes the current time of day to be displayed in the message window. This command is bound to Ctrl/T.

Example

```
Command:  SHOW TIME  Return
```

This command causes the current time to be displayed in the message window.

```
VAXcluster Console System 28-JUL-1988 12:09:45
```

SHUTDOWN

Executes an orderly shutdown of the VAXcluster Console System software.

Format

SHUTDOWN

Description

The SHUTDOWN command closes the log files, halts the VCS processes (such as the I/O Data Logger, Event Logger, and Data Scanner) and access applications, and notifies all interfaces. Typically, you issue the SHUTDOWN command only in special circumstances such as when you need to service the VCS host system upon recommendation of the system manager.

You must issue the SET PRIVILEGES command before you can enter the SHUTDOWN command.

Before you issue the SHUTDOWN command, you may want to ensure that you do not lose the information transmitted from each node to its console line. You also may need to control the nodes from an alternate console terminal.

The OPCOM utility usually directs information to both the console line and an operator's log file. However, if you issued the DCL command REPLY/NOLOG, OPCOM does not create the operator's log file. Before you issue the SHUTDOWN command, you may want to ensure that OPCOM is creating operator's log files for all nodes. Otherwise, the information produced by OPCOM will be lost when you issue the SHUTDOWN command.

To start up VCS after a shutdown, enter the following:

```
$  @SYS$STARTUP:VCS$STARTUP [Return]
```

When VCS$STARTUP.COM finishes executing, you can issue the VCS DCL commands to start up interactive VCS processes.

SHUTDOWN

Example

```
Command:  SET PRIVILEGES Return
Command:  SHUTDOWN Return
```

These commands stop VCS.

TWO_WINDOWS

Horizontally divides the single log window into two smaller log windows of equal size on the screen.

Format

TWO_WINDOWS

Description

Use the TWO_WINDOWS command to divide your screen display into two windows. You can view different parts of one log, or two different logs or buffers. For example, you can view a log in one window and the HELP buffer in the other window.

An asterisk (*) in the log window banner indicates the current window. Note that you can use the OTHER_WINDOW command to make the other window the current window.

Example

```
Command:  VIEW OMEGA [Return]
Command:  TWO_WINDOWS [Return]
Command:  HELP [Return]
```

View node OMEGA and change the screen display to two windows. When you enter the HELP command, the Monitor Interface displays help information in one window while you view log information for OMEGA in the other window.

UNDEFINE/KEY

Unbinds a text string from a key.

Format

UNDEFINE/KEY *(/qualifier . . .) key-name*

Parameters

key-name
is the name of the key or control character that is to be unbound from its current definition. Use the SHOW KEYS to determine the keys that are currently defined.

Qualifiers

/KEY
Is required with the UNDEFINE command to unbind a key definition.

/SHIFT_KEY
/NOSHIFT_KEY (D)
Undefine a key prefixed by a *shift* key. The key was defined by the DEFINE/KEY/SHIFT_KEY command.

The default is /NOSHIFT_KEY.

Description

The UNDEFINE/KEY command removes the binding of a function to a key. This action is in effect for the current monitor session only. To preserve the change, use the SAVE command to create a new monitor section file.

Example

```
Command:  SHOW KEYS [Return]
Command:  UNDEFINE/KEY CTRL_Y_KEY [Return]
Command:  SAVE USER:PRIVATE_SECTION.FILE [Return]
```

The example demonstrates how to display defined keys, undefine Ctrl/Y, and save the change in the USER:PRIVATE_SECTION.FILE file.

VIEW

Displays information from the specified buffer in the current log window.

Format

VIEW { *node-name* } { */qualifier* }

Parameters

node-name
is the name that the system manager assigns to a node in the VCS configuration file. You can also use a previously defined logical name. You can abbreviate node names, but you cannot abbreviate logical names. When you specify a list of node names or logical names, separate the names with commas.

This must be one of the nodes that you are currently monitoring in the Monitor Interface. Use the SHOW NODES command to determine the list of the node names you may use in this command.

Qualifiers

/ACTIVE
indicates the log buffer of the first node, leftmost one, listed in the event summary window is to be displayed.

/CONTEXT
indicates that the context of the current event, in the Event Log window that you are viewing, is to be displayed.

/EVENT
indicates that the event log buffer is to be displayed.

/HELP
indicates that the current information in the help buffer is to be displayed. The information in the help buffer is generated when you enter the HELP command. If you have not executed the HELP command during your current session, the help buffer is empty.

/MESSAGE
indicates that the message buffer is to be displayed. The message buffer contains all of the error and system messages that were either generated by your activities during the current session or received from the system.

/NEXT
indicates that the log buffer of the next node from the VCS MONITOR command list is to be displayed.

Description

The VIEW command displays requested information from the specified buffer in the current log window. The requested buffer can be either a node's log buffer, the event log buffer, the event context buffer, the message buffer, or the help buffer.

VIEW/ACTIVE enables you to view a node that has transmitted log information since you last viewed it. Any node in this state will have an event listed for it in the event summary window. The events are listed in order of priority, with the highest priority event at the left. VIEW/ACTIVE removes the leftmost event from the event summary window, and displays the associated node buffer in the current log window. Subsequent VIEW/ACTIVE commands enable you to view, in turn, all nodes with new log information.

When you began the Monitor Interface session using the VCS MONITOR command, you specified a list of nodes whose buffered log information would be available to you during the current session. For example, when you began your current Monitor Interface session, you typed:

```
$ VCS MONITOR ALPHA, BETA, OMEGA Return
```

VIEW

VIEW/NEXT selects the next node named in the list and displays current log information for that node. For example, if your current node is OMEGA, then VIEW/NEXT displays the log of ALPHA. Subsequent VIEW/NEXT commands enable you to view, in turn, every node available to this session.

VIEW/CONTEXT enables you to view the Event Context window. The text of the event (as seen in the Event Log window) is included in the status line of the event context window. This text is delimited by braces ({ }).

If your previous view of a log buffer was at any point other than the most recent data, the next view of that buffer will return to that same point. Such a static view of the buffer is attained through the GOTO FIRST, SCROLL UP, FIND/REVERSE or PREVIOUS SCREEN commands. If your previous view of a buffer was the most recent data, or if the buffer was not previously viewed, the next view of that buffer will be the most recent data. Such a dynamic view is restored through the GOTO LAST command or sufficient SCROLL DOWN or NEXT SCREEN commands.

Examples

1 `Command: VIEW OMEGA` Return

This command enables you to view the log buffer for OMEGA.

2 `Command: VIEW/MESSAGE` Return

This command views a list of all the messages you have received during your current session.

WAIT

Prevents the Monitor Interface from accepting user commands until the specified period of time has elapsed.

Format

WAIT *delta-time*

Parameters

delta-time
is the time interval to wait. A delta time is an offset from the current time to a time in the future. The format for delta time is:

```
hh:mm:ss.cc
```

Specify hours, minutes, seconds, and hundredths of seconds, but not days. You must begin the delta time with the number of hours and not a colon, even if the number of hours is zero. For more information on delta time, refer to the *VMS DCL Concepts Manual*.

Description

The WAIT command prevents the Monitor Interface from accepting user commands until the specified period of time has elapsed. You can interrupt this input suspension by pressing the escape key. The default escape key is Ctrl/G. See the SET ESCAPE command for more information.

The WAIT command is an ideal way to set the tempo for the execution of a command file. For example, when you compose the contents of a command file that contains several OUTPUT commands, you may want to insert the WAIT commands between the appropriate OUTPUT commands to allow the serviced system time to process the text strings. For example, the HSC device may take up to a second to response to the Ctrl/C entered at its console. The Monitor Interface may execute a series of OUTPUT commands so rapidly that the HSC console port is overrun.

You may want to create such a series of OUTPUT commands as follows:

```
OUTPUT/NOTERMINATE HSC001 Ctrl/V Ctrl/C
WAIT 00:00:02.00
OUTPUT HSC001 "SHOW DISKS"
```

Example

```
Command:  WAIT 00:01:30 Return
```

This example sets a waiting interval of 1 minute and 30 seconds.

WRITE

Copies the contents of the current buffer to a specified file.

Format

WRITE *file-name (/qualifier)*

Parameters

file-name
is the file to contain the current buffer contents.

Qualifiers

/TIME_STAMP (D)
/NOTIME_STAMP
indicates whether the time stamps are to be included in the data to be written to the file. The time stamps exist in the node log buffers only. When you write the contents of other buffers such as help or message to a file, this qualifier is ignored.

The default is /TIME_STAMP.

Description

Use the WRITE command to copy the contents of the current buffer to a file. An asterisk (*) in the log window banner indicates the current window.

Example

```
Command:  VIEW BETA [Return]
Command:  WRITE SYSTEM_BETA.LOG [Return]
```

This command displays the current log for BETA, then writes the contents of the current log buffer for BETA to the file, SYSTEM_BETA.LOG.

7

Summary of Monitor Interface Tasks

This chapter presents a summary of user tasks with a brief explanation of their related commands. The subsections that follow are in the same order as the tasks in Table 7–1. **The commands listed in this chapter are not presented here in full detail. For more information on a specific DCL command, see Chapter 5 or the online HELP. For more information on a specific Monitor Interface command, see Chapter 6 or the online HELP.** Note that all DCL-level commands in these tables are preceded by **$ VCS**; all Monitor Interface commands are not.

Table 7–1 summarizes Monitor Interface user tasks and the commands necessary to accomplish the tasks.

Table 7–1 Monitor Interface Tasks and Commands

User Task	Monitor Interface Command	DCL Command
Entering and exiting the Monitor Interface	EXIT	$ VCS MONITOR
Getting help	HELP	$ HELP VCS
Communicating with nodes	SELECT CONNECT OUTPUT	$ VCS CONNECT
Viewing current console log windows	VIEW CYCLE	

(continued on next page)

Table 7–1 (Cont.) Monitor Interface Tasks and Commands

User Task	Monitor Interface Command	DCL Command
Viewing events	VIEW/EVENT NEXT EVENT PREVIOUS EVENT	
Viewing event context	VIEW/CONTEXT NEXT EVENT PREVIOUS EVENT	
Controlling the window display	GOTO FIRST GOTO LAST NEXT SCREEN PREVIOUS SCREEN REFRESH SCROLL DOWN SCROLL UP SHIFT LEFT SHIFT RIGHT ONE_WINDOW OTHER_WINDOW SET WIDTH TWO_WINDOWS	
Searching for information in a buffer	FIND/FORWARD FIND/REVERSE	
Showing information about the current Monitor Interface session	SHOW DIRECTORY SHOW MONITOR SHOW NODES SHOW KEYS SHOW TIME	
Reviewing log data	REVIEW REVIEW/EVENT	$ VCS REVIEW $ VCS REVIEW/EVENT
Copying a buffer into a file	WRITE	
Viewing the messages received in the current Monitor Interface session	VIEW/MESSAGE	
Modifying key definitions	DEFINE/KEY UNDEFINE/KEY SET SHIFT_KEY SAVE	

(continued on next page)

Table 7–1 (Cont.) Monitor Interface Tasks and Commands

User Task	Monitor Interface Command	DCL Command
Tailoring your Monitor Interface environment	SET BELL SET BREAK SET ESCAPE SET MOUSE SET TIME_STAMP	

7.1 Entering and Exiting the VCS Monitor Interface

Options	Commands
Entering the Monitor Interface	$ VCS MONITOR
Exiting the Monitor Interface	EXIT

Description

Start a VCS Monitor Interface session by entering the $ VCS MONITOR command and specifying the names of the nodes you want to monitor. The default is VCS$ALL_NODES, which is a logical name that specifies all the nodes in the current VCS configuration.

To execute a command file that contains Monitor Interface commands, use the $ VCS MONITOR command with the /COMMAND qualifier. Use the /SECTION qualifier to load a TPU section file containing your custom key definitions for the current session. The /NOEVENT qualifier specifies that you do not want the event indicators displayed and you do not want the Event Log window present. The /ACCESS_MODES=MONITOR qualifier indicates that the Monitor Interface will be a display-only (non-interactive) session.

To end a Monitor Interface session, enter EXIT.

7.2 Getting Help

Options	Commands
Get HELP on VCS commands	HELP $ HELP VCS
Exit HELP	Press Ctrl/Z
To remove the HELP display from your screen in the Monitor Interface.	View another window
To view HELP information from the most recent HELP command issued in the Monitor Interface	VIEW/HELP

Description

To display help information on the DCL-level VCS commands, you must enter the $ HELP VCS command at the DCL prompt. To display help information on the VCS Monitor Interface commands, enter HELP at the **Command:** prompt within the Monitor Interface. The HELP command displays a list of commands for which information is available, and displays the **Topic?** prompt on the VCS command line. At the **Topic?** prompt:

- Enter one of the topic names from the list.
- Enter HELP DESCRIPTION for more information about how to use the HELP facility.
- Press RETURN one or more times or press Ctrl/Z to exit from HELP.
- Enter a question mark (?) to display all available topics.
- Enter a topic name followed by an ellipsis (. . .) to display all the information on a topic. For example, entering CONNECT . . . at the **Topic?** prompt displays all the information available for the CONNECT command.

7.3 Communicating with Nodes

The Monitor Interface provides three ways to communicate with the serviced nodes:

Options	Commands
Connect your terminal to a single node while retaining the VCS display and monitoring events.	SELECT
Connect your terminal to a single node with the node's display replacing the VCS display.	CONNECT $ VCS CONNECT
Send commands to one or more nodes while retaining the VCS display and monitoring events.	OUTPUT
Exit the CONNECT, $ VCS CONNECT, or SELECT session.	Press the escape character.

Description

Use the SELECT command to communicate with a node while retaining the Monitor Interface display on your screen and continuing to monitor events for the serviced nodes. When you issue the SELECT command, you can specify a node name, /ACTIVE, or /NEXT, or you can default to the currently viewed node. When you issue the SELECT command, the cursor moves into the log window, and the event indicators for the selected node are no longer displayed in the event summary window. The SELECT command notifies you of the escape character you must use to exit the SELECT mode.

The CONNECT and $ VCS CONNECT commands connect your terminal to a node. The node controls your full terminal screen, replacing the Monitor Interface display. Use these commands when you want to run a program that requires full-screen control, or when you want the full terminal screen for your work. When you issue the CONNECT command, you can specify a node name, /ACTIVE or /NEXT, or you can default to the currently viewed node. When you issue the $ VCS CONNECT command, you must specify a node name. Both the CONNECT and $ VCS CONNECT commands notify you of the escape character you must use to exit CONNECT mode.

Issue the OUTPUT command to send a single command to more than one node at a time, while retaining the Monitor Interface display on your screen and continuing to monitor events for the serviced nodes. You do not have to enclose the command in quotation marks, although you may include them. You can send a break signal to one or more nodes using the /BREAK qualifier to the OUTPUT command.

7.4 Viewing Current Console Logs

Options	Commands
Display current log information in the log window.	VIEW
Continuously view each log window, one at a time.	CYCLE

Description

The VCS Monitor Interface maintains a series of log buffers that store console data and messages from the various nodes. Issue the VIEW command to inspect the latest contents of these buffers. You can specify a node name, /NEXT or /ACTIVE.

The CYCLE command automatically and continually displays the contents of each of the console logs of the nodes that you are monitoring. This command is equal to entering a series of VIEW/NEXT and WAIT commands.

7.5 Viewing Events

Option	Commands
View the Event Buffer	VIEW/EVENT
Change the current event to be the previous event	PREVIOUS EVENT
Change the current event to be the next event	NEXT EVENT

Description

You are notified of events in the Event Summary window. The Monitor Interface displays only the event name of the highest priority event for each node in the Event Summary window. The Monitor Interface also stores the events in the Event buffer. You can view the Event buffer using the VIEW/EVENT command. The current event is highlighted according to its priority.

Using the PREVIOUS EVENT and NEXT EVENT commands, you can change the current event to be the event prior to the current event or the event after the current event.

7.6 Viewing Event Context

Options	Commands
View the Event Context buffer	VIEW/CONTEXT
View the context of the previous event	PREVIOUS EVENT
View the context of the next event	NEXT EVENT

Description

You are notified of events in the Event Summary window. The Monitor Interface displays only the event name of the highest priority event for each node in the Event Summary window. You can view the context of the current event (in the Event Context window) by entering the VIEW/CONTEXT command.

Once you are viewing the Event Context window, you can enter the PREVIOUS EVENT or NEXT EVENT command. These commands change the current event to be the event prior to the current event or the event after the current event, and update the Event Context window to contain the context of that event.

7.7 Controlling the Window Display

Options	Commands
Display the next screen of information.	NEXT SCREEN or Next Screen or KP0 on numeric keypads
Display the previous screen of information.	PREVIOUS SCREEN or Prev Screen or the period (.) key on numeric keypads
Restore the screen and eliminate extraneous characters and messages.	REFRESH or Ctrl/W
Move the current window toward the most recent data, one line at a time.	SCROLL DOWN or KP2 on numeric keypads
Move the current window toward the oldest data, one line at a time.	SCROLL UP or KP5 on numeric keypads
Move the current window to the left.	SHIFT LEFT or KP1 on numeric keypads
Move the current window to the right.	SHIFT RIGHT or KP3 on numeric keypads
Display one window on your screen.	ONE_WINDOW
Display two windows on your screen.	TWO_WINDOWS

Options	Commands
Change the current window from one window to the other window (when you have two windows displayed on your screen).	OTHER_WINDOW
Set the width of your terminal display.	SET WIDTH

7.8 Searching for Information

Options	Commands
Search a buffer from the current location to the next location of a string.	FIND/FORWARD
Search a buffer from the current location to the previous location of a string.	FIND/REVERSE
Display the oldest information for the current buffer.	GOTO FIRST
Display the newest information for the current buffer.	GOTO LAST

Description

If you issue the FIND command and do not specify /FORWARD or /REVERSE, the Monitor Interface searches in the last direction you specified with a FIND command. The default direction before any FIND command is issued is /FORWARD. If you do not specify a string with the FIND command, the Monitor Interface searches for the last string that you specified with this command.

You can search any buffer with the FIND command, as long as the buffer is in the current window.

The GOTO FIRST command moves the current window to the first or oldest information in the buffer. GOTO LAST moves the current window to the last or most recent information in the buffer.

7.9 Showing Information About the Current Monitor Interface Session

Options	Commands
Display information about the console and event logs available on disk.	SHOW DIRECTORY
Display information about the monitor session attributes and parameters.	SHOW MONITOR
Display information about the nodes available to the current session.	SHOW NODES
Display information about the key definitions in the current session.	SHOW KEYS
Show the current time of day.	SHOW TIME

7.10 Reviewing Log Data

Options	Commands
Review console logs	REVIEW $ VCS REVIEW
Review event logs	REVIEW/EVENT $ VCS REVIEW/EVENT
Redisplay the previous review information	REVIEW/PREVIOUS
Write the reviewed data into an output file	WRITE $ VCS REVIEW/OUTPUT

Description

You can access log data stored in the VCS console and event log files by issuing the REVIEW or $ VCS REVIEW command. By default, you will review console log data. Specify /EVENT to review event log data. You can specify a single node or a list of nodes. Specify a period of time for which you want to review data by using the /SINCE=date-time and /BEFORE=date-time qualifiers.

Enter the REVIEW/PREVIOUS command to redisplay the same information that the last REVIEW command displayed.

To write the review data into an output file, issue the WRITE command, after the REVIEW command completes, if you are in the Monitor Interface. From DCL, use the /OUTPUT switch.

7.11 Copying a Buffer into a File

Option	Command
Copy the contents of the current buffer to a file.	WRITE

Description

The WRITE command copies the contents of the current buffer into a file with the name you specify. You can write any buffer to a file.

7.12 Viewing the Messages Received in the Current Monitor Interface Session

Option	Command
Display the buffer containing messages you receive during your current Monitor Interface session.	VIEW/MESSAGE

Description

The Monitor Interface stores VCS and system broadcasted messages that you receive during your current session. You can display these messages by issuing the VIEW/MESSAGE command.

7.13 Modifying Key Definitions

Options	Commands
Assign a command to a key.	DEFINE/KEY
Undefine a key definition.	UNDEFINE/KEY
Define the shift key to be used when defining keys.	SET SHIFT_KEY
Preserve your key definitions in a file.	SAVE

Description

Assign a string to a key by using the DEFINE/KEY command. Assign a string to a shifted key with the DEFINE/KEY/SHIFT_KEY command. Issue the SET SHIFT_KEY command to identify the shift key. To remove the binding of a function to a key, use the UNDEFINE/KEY. Use UNDEFINE/KEY/SHIFT_KEY to remove the binding of a function to a shifted key. Use the /TERMINATE qualifier to terminate the DEFINE/KEY command string with the carriage-return line-feed characters. Issue the SAVE command to create a file with all the key definitions you defined during your session. To start a new session and use these key definitions,

issue the $ VCS MONITOR/SECTION=file-name command where file-name specifies the file that contains your saved key definitions.

7.14 Tailoring Your Monitor Interface Environment

Use the following SET commands to tailor your VCS environment:

Options	Commands
Enable or disable the bell tone associated with the events displayed in the event window.	SET BELL
Define a key that sends a break signal to nodes when you use the SELECT or CONNECT commands.	SET BREAK
Define a key that exits you from CONNECT or SELECT commands.	SET ESCAPE
Enable the VAXstation mouse for use as a locator device.	SET MOUSE
Control whether the time stamp is displayed at the beginning of each line of a log window display.	SET TIME_STAMP

8

Creating Your Own Icons

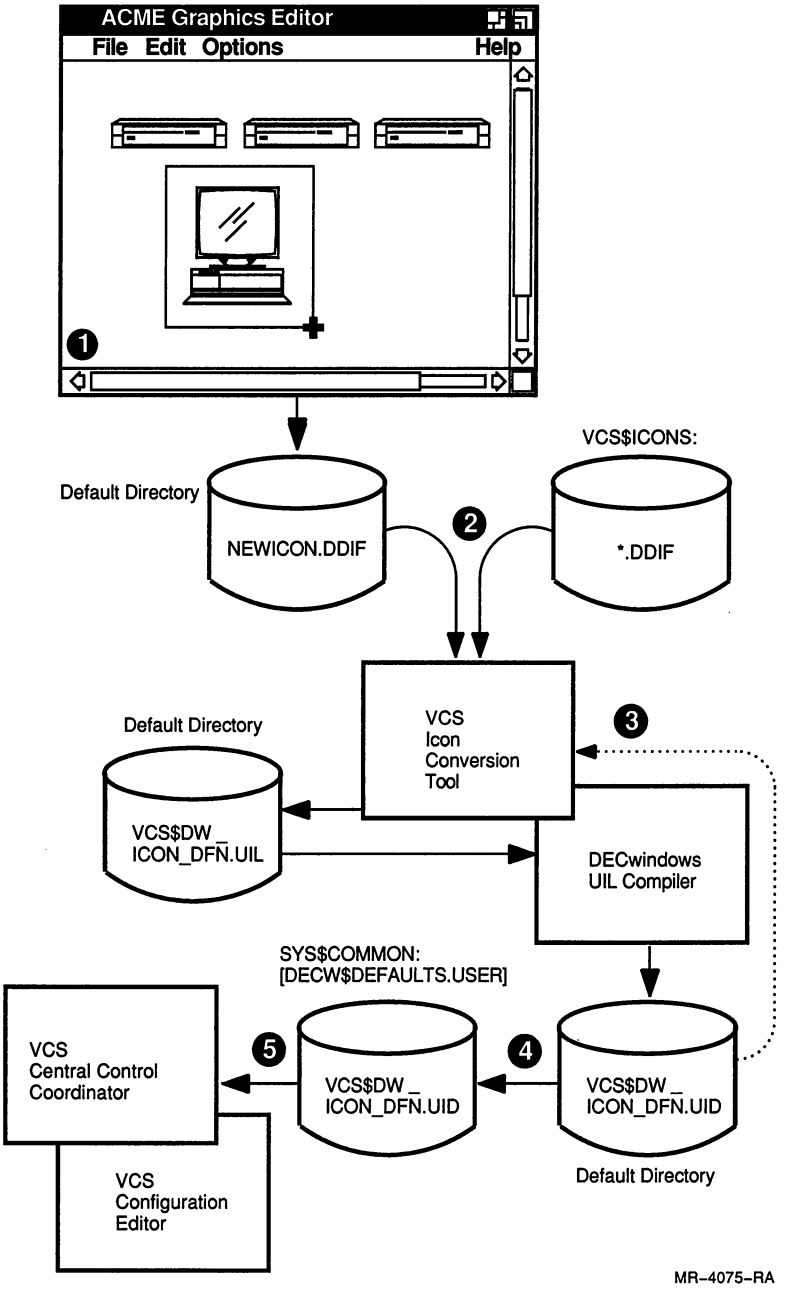
The VCS CONVERT /ICON command allows you to create custom icons for the VCS Central Control Coordinator. The five major steps necessary to create your own icons are illustrated in Figure 8–1. Those steps are:

1. Drawing an icon.
2. Converting the icon from a DDIF file into a DECwindows UIL (User Interface Language) and then a UID (User Interface Description) file.
3. Reviewing the icon.
4. Making the new icon available to VCS.
5. Using the new icon.

Restrictions *In order to create your own icons, your VCS host system must have the UIL compiler that is part of the DECwindows programming kit.*

You cannot substitute any other icon for the VCS host icon.

Figure 8–1 Creating Your Own Icons



8.1 Creating an Icon Picture

The first step in creating your own icon is drawing or capturing the picture. This picture must be in the Digital Document Interchange Format (DDIF).

8.1.1 Drawing an Icon Picture

To draw the picture of the icon, you can use any graphics editor that can save a monochrome image in the DECwindows DDIF format. The DECwindows Paint utility is a suitable graphics editor.

The picture you draw can be any shape that has an unbroken outline enclosing an inner region. The enclosed inner region is necessary so VCS can distinguish the interior of the icon from the screen background when the Central Control Coordinator changes the icon color.

Your icon must be less than 250 pixels wide and 250 pixels high. Keep in mind how many icons you will have on your display at one time when deciding on the size of the icon.

8.1.2 Capturing an Icon Picture

An alternative to drawing the icon is capturing an existing figure displayed on the screen by any DECwindows application. This can be done by using the DECwindows Session Manager **Print Screen** menu item **Capture Portion of Screen**.

Before capturing an icon off the screen, remember to set up the **Customize** menu **Print Screen** information to produce the image in DDIF format.

To capture a portion of the screen, perform the following steps:

1. Move the pointer to the DECwindows Session Manager Print Screen menu. Press and hold MB1.
2. Drag the pointer down. Choose the Capture Portion of Screen menu item and release MB1. A dialog box is displayed.
3. Click on the OK button to capture the image. Click on the Cancel button to cancel the operation. The dialog box also allows you the option of changing the name of the file that contains the image.
4. Your workstation displays a cross-like pointer. Move the pointer to the area of the screen you want to capture.

5 Press and hold MB1. Move the pointer to create an outline that will enclose the portion of the screen you want to capture.

6 Release MB1. The portion of the screen within the outline will be captured to the file you chose.

When capturing the icon image, be sure to leave enough room between the image outline and the capture outline so that you are not cropping any of your icon. Excess background between the image outline and the capture outline will be removed by the VCS icon conversion tool.

8.2 Converting the Image

Once you have a DDIF file, the next step is to convert the picture into a DECwindows UID file which can be used by the VCS Configuration Editor and Central Control Coordinator.

VCS icon conversion is a two-step process. First, the VCS CONVERT /ICON command accepts DDIF format files as input and converts them into a UIL file. By default, a file called VCS$DW_ICON_DFN.UIL is created in the default directory. This file contains the descriptions of the icons as DECwindows UIL pixmaps.

Then the DECwindows UIL compiler is automatically invoked to compile this newly created UIL file. The result is placed in the file VCS$DW_ICON_DFN.UID in the default directory.

Sometimes you may create or capture an image in which the foreground and background colors are inverted. The VCS icon conversion tool defines the lighter color of an image as the background and the darker color as the foreground. You can invert that definition with the /INVERT qualifier at conversion time.

For example, you may have one new icon that requires an inverted bitmap, and you want to add it to the icons that are supplied with VCS. To do that, you first convert all the icons that do not require an inverted bitmap. Enter the following command:

```
$ VCS CONVERT /ICON VCS$ICONS:*.DDIF Return
```

The files VCS$DW_ICON_DFN.UIL and VCS$DW_ICON_DFN.UID are created in the default directory.

To convert the one icon that requires the inverted bitmap, enter the following command:

```
$ VCS CONVERT /ICON /APPEND /INVERT NEWICON.DDIF Return
```

This second step converts the inverted image and appends the result to the existing VCS$DW_ICON_DFN.UIL file in the default directory. Then the UIL compiler creates VCS$DW_ICON_DFN.UID in the default directory. VCS$DW_ICON_DFN.UID now contains all the icons converted in the both commands.

The only way to remove an icon from from the UID file is to reconvert all the icons you want to retain, excluding the icon you want to remove.

8.3 Reviewing the Icon

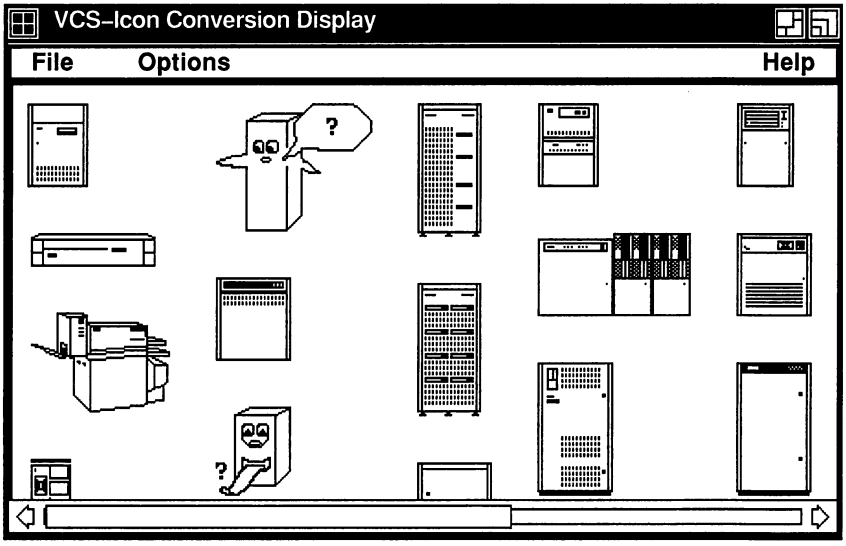
Use the /REVIEW qualifier to display the newly created icon. Reviewing the icon lets you compare your new icon with existing icons. If you still want to change the new icon, you can now touch up the DDIF file with your graphics editor, then convert and review the touched up image as many times as you want.

When you use the /REVIEW qualifier without converting any input file (as in the preceding section), the VCS icon conversion tool reviews the contents of the file VCS$DW_ICON_DFN.UID. To review a different UID icon file, use the /UID_FILE qualifier, as described in Chapter 5.

For example, to display all the icons converted in the previous section, enter the following command:

```
$ VCS CONVERT /ICON /REVIEW Return
```

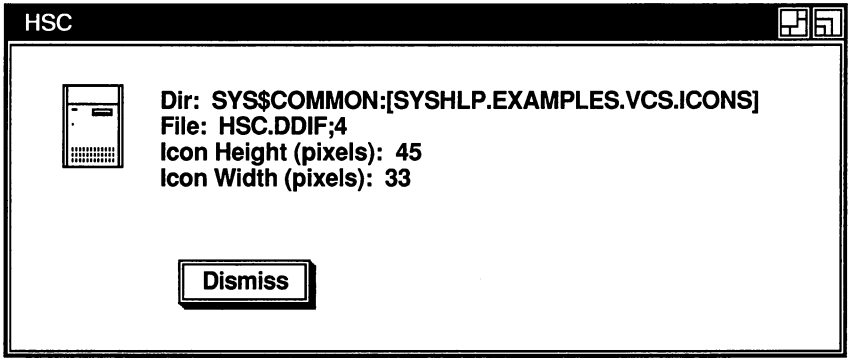
VCS displays the Icon Conversion Display window.



In order to compare icons easily, you can move the icons around within the window. To move an icon, move the pointer to the icon and drag it by pressing and holding MB1.

8.3.1 Getting Information About an Icon

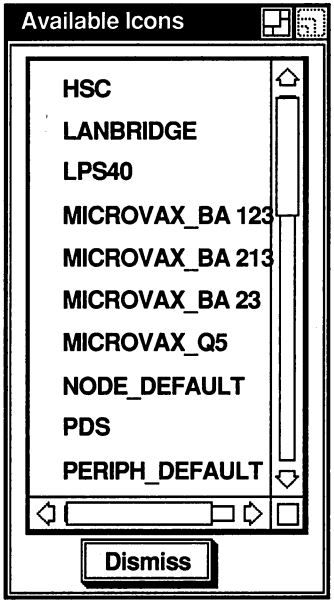
To get information about an icon, click MB2 on the icon.



An icon information box is displayed. The box shows the DDIF file name and the directory where the icon was converted, as well as the dimensions of the icon in pixels. Click on Dismiss to close the icon information box.

Options
Icon Names...

Another way to get information about an icon is to select the Icon Names... menu item from the Options menu. The icon review displays a list box.



Click on the name of the icon to display the icon information box. Click on Dismiss to close the Available Icons list box.

8.3.2 Icon Names

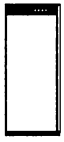
When you convert a new icon, it is given the same name as the DDIF file from which it was converted. That name is used by the Icon Conversion Display and by the Configuration Editor.

DDIF file names that start with a digit are given icon names that begin with an underscore (_). DDIF file names longer than 18 characters are truncated. Dashes (-) in DDIF file names are changed to underscores (_).

8.3.3 Icons Supplied by VCS

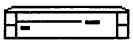
The following list shows the icons that are supplied by VCS.

EXPANSION

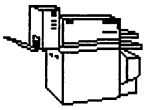


HSC

LANBRIDGE



LPS40



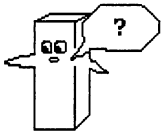
MICROVAX_BA123

MICROVAX_BA213

MICROVAX_BA23

MICROVAX_Q5

NODE_DEFAULT



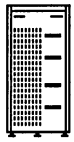
PDS

PERIPH_DEFAULT

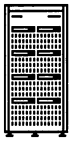
PRINTER

RA81

SA482

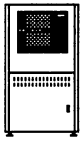


SA600

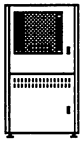


STARCOUPLER

TA78

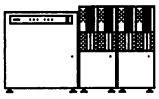


TA79

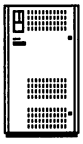


TA81

TA90

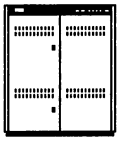


VAX6000

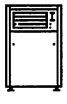


VAX750

VAX780

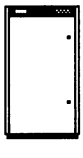


VAX8200C1

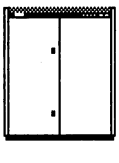


VAX8200C2

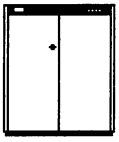
VAX8500



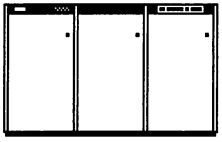
VAX8600



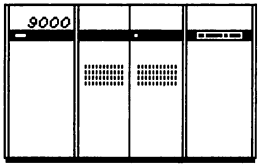
VAX8800



VAX9000



VAX9410



VAXSTATION3100

VAXSTATION3500

8.3.4 Exiting from the Icon Review

File
Quit

Choose the Quit menu item from the File menu to exit from the icon review and close the Icon Conversion Display window.

8.4 Using the Icon

After you are satisfied with the newly created icon, you must copy the new file VCS$DW_ICON_DFN.UID to the directory SYS$COMMON:[DECW$DEFAULTS.USER] where the VCS DECwindows interfaces can access the new set of icons.

At this time, you may also want to save the icon DDIF file in the directory VCS$ICONS.

From now on, each time you invoke the VCS Configuration Editor or Central Control Coordinator, the new icon is available on the icon selection menu.

Note *If you invoke the VCS DECwindows interfaces and unexpectedly see the default node or default peripheral icons, you may have left out one or more icon DDIF files when you converted your icons.*

9

Writing Access Interface Applications

VCS enables you to develop your own Access Interface application software to enhance the processing of VCS and console activities. You develop your application software as a collection of *callback routines*. These callback routines must be named and must adhere to a set of calling conventions as described in this chapter. When VCS detects a scan event on a console line, or when a VCS specific event occurs (for example, a VCS reconfiguration or a user activation of an interface), the Access Interface executes an associated callback routine. VCS provides the callback routines with the information about the event (for example, the time stamp and the node name) allowing possible additional processing to handle the event.

This chapter describes:

- An overview of the callback routines
- Guidelines for designing the user application
- How to create and organize the user code
- How to link and execute the user code
- How to debug the user code
- The callback routine reference

To allow you to develop Access Interface applications quickly, the product software kit contains a number of sample code modules written in VAX C. These are complete and working examples of how to develop an Access Interface application. These files are placed in the directory VCS$EXAMPLES during the software installation.

9.1 Overview of the Callback Routines

The system programmer must provide the callback routines listed in Table 9–1 to interact with the Access Interface. These routines are not expected to return any value to the Access Interface. The application must handle any exception condition occurring during the execution of these routines.

Table 9–1 Access Interface Callback Routines

Callback Routine	Type	Description
VCS$USER_INITIALIZE	Required	Initializes data storage, input and output devices, and the environment. This routine is called once at the beginning of the process. This routine processes the user-defined command line qualifiers. You can set up exit handlers in this routine which are invoked as part of the image rundown if the image is terminated abnormally.
VCS$USER_MAINLOOP	Optional	Provides access to the DECwindows environment. This routine is called only once at the beginning of the process and only if your application intends to provide a DECwindows interface. This routine is expected to register a DECwindows work procedure given by the Access Interface and to pass program control on to the DECwindows MAIN LOOP intrinsic routine.
VCS$USER_RECONFIGURE	Optional	Processes the VCS reconfiguration event. This routine is called each time the RECONFIGURE command is issued.
VCS$USER_SCAN_EVENT	Optional	Processes console event indicators detected by and relayed from the Data Scanner. This routine is called each time any event indicator is detected on a console line.
VCS$USER_SHUTDOWN	Required	Terminates processing. This routine is called once at the end of processing.
VCS$USER_VCS_EVENT	Optional	Processes VCS events triggered by various VCS specific activities and conditions which affect the operations of VCS on the host system. This routine is called each time such an event is detected.

The callback routines can be written in any programming language that supports the VMS procedure calling standards. However, the required symbols to be used in the callback routines are defined only for some of the most common programming languages as listed in Table 9–2.

Table 9–2 VCS Common Source Files

Language	Common Source File
BLISS	VCS$DEFINE.R32
VAX C	VCS$DEFINE.H
MACRO	VCS$DEFINE.MAR
PASCAL	VCS$DEFINE.PAS
FORTRAN	VCS$DEFINE.FOR

9.2 Creating User Code

Use your favorite text editor to create and modify the source program. This section uses VAX C to illustrate a typical organization of the callback routines.

The file VCS$APPLICATION.C, which can be found in the directory VCS$EXAMPLES, contains the following skeleton code. Annotations for the callout numbers follow Example 9–1.

Example 9–1 Sample Application Code

```
#module VCS$APPLICATION "Access Interface Example  -0001"
#include <decw$include:DwtAppl> ❶
#include <descrip>
#include <stdio>
#include "vcs$define" ❷

void vcs$user_initialize (itmlst) ❸
    struct pairs {                      /* item list pairs */
        unsigned long code;             /* argument code */
        variant_union {                 /* argument value */
            unsigned long count;        /* number of systems on command line */
            struct dsc$descriptor_s *name; /* system name descriptor */
            struct dsc$descriptor_s *display; /* display name descriptor */
            struct pairs *systems;      /* pointer to item list of systems */
        } value;
    } itmlst[];
{
    unsigned long i;

    for (i = 0; itmlst[i].code != VCS$K_END_OF_LIST; i++) ❹

        switch (itmlst[i].code) {
```

(continued on next page)

Example 9–1 (Cont.) Sample Application Code

```
            case VCS$K_SYSTEM_COUNT:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_SYSTEM_LIST:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_DECW_DISPLAY: ❺
                Your logic goes in here
                .
                .
                .
                break;

            default:
                fprintf (stderr, "Unknown argument: 0X%8X\n", itmlst[i].code);
                break;
        }
    Your logic goes in here ❻
    .
    .
    .

    return;                             /* done */

}                                       /* end of vcs$user_initialize */
void vcs$user_mainloop (itmlst) ❼
    struct pairs {                      /* item list pairs */
        unsigned long code;             /* argument code */
        variant_union {                 /* argument value */
            int (*work_procedure)(); /* work procedure */
            unsigned long address; /* address of work procedure */
        } value;
    } itmlst[];
{

    XtWorkProcId procedure;
    unsigned long i;

    for (i = 0; itmlst[i].code != VCS$K_END_OF_LIST; i++) ❽

        switch (itmlst[i].code) {

            case VCS$K_WORK_ADDRESS:
                Your logic goes in here
                .
                .
                .
                procedure = XtAddWorkProc (itmlst[i].work_procedure, 0); ❾
                .
                .
                .
                break;
```

(continued on next page)

Example 9–1 (Cont.) Sample Application Code

```
            default:
                fprintf (stderr, "Unknown argument: 0X%8X\n", itmlst[i].code);
                break;
        }
    Your logic goes in here
    .
    .
    .
    XtMainLoop(); (10)
}                                       /* end of vcs$user_mainloop */

void vcs$user_shutdown () (11)
{

    Your logic goes in here (12)
    .
    .
    .

    return;                             /* done */
}                                       /* end of vcs$user_shutdown */

void vcs$user_reconfigure (itmlst) (13)
    struct pairs {                      /* item list pairs */
        unsigned long code;             /* argument code */
        variant_union {                 /* argument value */
            unsigned long count;        /* number of systems in the new configuration */
            struct dsc$descriptor_s *name; /* system name descriptor */
            struct pairs *systems;      /* pointer to item list of systems */
        } value;
    } itmlst[];
{
    unsigned long i;

    for (i = 0; itmlst[i].code != VCS$K_END_OF_LIST; i++) (14)

        switch (itmlst[i].code) {

            case VCS$K_SYSTEM_COUNT:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_SYSTEM_LIST:
                Your logic goes in here
                .
                .
                .
                break;

            default:
                fprintf (stderr, "Unknown argument: 0X%8X\n", itmlst[i].code);
                break;
        }
    Your logic goes in here (15)
    .
    .
    .
```

(continued on next page)

Example 9–1 (Cont.) Sample Application Code

```
    return;                                 /* done */

}                                           /* end of vcs$user_reconfigure */

void vcs$user_scan_event (itmlst) 16
    struct {
        unsigned long code;
        variant_union {
            struct dsc$descriptor *system; /* system name */
            struct dsc$descriptor *event; /* event name abbreviation */
            struct dsc$descriptor *event_name; /* event name in full */
            struct dsc$descriptor *text; /* console text line */
            unsigned long match;        /* offset and length of matched text */
            unsigned long priority;     /* event priority */
            unsigned long *time;        /* event time stamp */
            unsigned long *context_timestamp; /* context time stamp */
            unsigned long context_count;/* number of context text lines */
            struct dsc$descriptor *info;/* user information */
        } value;
    } itmlst[];
{
    unsigned long i;

    for (i = 0; itmlst[i].code != VCS$K_END_OF_LIST; i++) 17

        switch (itmlst[i].code) {

            case VCS$K_SYSTEM_NAME:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_EVENT_ABBREVIATION:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_EVENT_NAME:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_SCAN_TEXT:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_SCAN_MATCH:
                Your logic goes in here
                .
                .
                .
                break;
```

(continued on next page)

Example 9–1 (Cont.) Sample Application Code

```
            case VCS$K_PRIORITY:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_TIME:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_CONTEXT_START:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_CONTEXT_COUNT:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_INFORMATION:
                Your logic goes in here
                .
                .
                .
                break;

            default:
                fprintf (stderr, "Unknown argument: 0X%8X\n", itmlst[i].code);
                break;
        }
    Your logic goes in here (18)
    .
    .
    .

    return;                                 /* done */

}                                           /* end of vcs$user_scan_event */

void vcs$user_vcs_event (itmlst) (19)
    struct {
        unsigned long code;
        variant_union {
            unsigned long *time;            /* event time stamp */
            struct dsc$descriptor *system; /* system name */
            unsigned long event;            /* event code */
            unsigned long pid;              /* user process id */
            struct dsc$descriptor *text;/* event text message */
        } value;
    } itmlst[];
{
    unsigned long i;

    for (i = 0; itmlst[i].code != VCS$K_END_OF_LIST; i++) (20)
```

(continued on next page)

Example 9–1 (Cont.) Sample Application Code

```
        switch (itmlst[i].code) {

            case VCS$K_TIME:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_SYSTEM_NAME:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_EVENT_CODE:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_USER_PID:
                Your logic goes in here
                .
                .
                .
                break;

            case VCS$K_EVENT_TEXT:
                Your logic goes in here
                .
                .
                .
                break;

            default:
                fprintf (stderr, "Unknown argument: 0X%8X\n", itmlst[i].code);
                break;
        }

    Your logic goes in here (21)
    .
    .
    .

    return;                             /* done */

}                                       /* end of vcs$user_vcs_event */
```

The following text elaborates on the callout numbers in Example 9–1 and provides additional hints for writing the Access Interface application code.

Prologue.

❶ The file DECW$INCLUDE:DWTAPPL.H is needed only if your application provides a DECwindows interface. Specify other DECwindows definition files as your application needs them here.

❷ The file VCS$DEFINE.H is installed in SYS$LIBRARY and contains many symbol definitions that you will need in your application code. Other symbol definition header files for other programming languages are listed in Table 9–2 and can also be found in SYS$LIBRARY.

Routine VCS$USER_INITIALIZE.

❸ The callback routine VCS$USER_INITIALIZE processes the passed arguments which specify the count and name of the systems on the command line. It also processes additional command line data directly, which may vary across applications. This callback routine is required.

❹ This loop passes through the entire argument block to retrieve the list of nodes that the user specified in the DCL command VCS ACCESS.

❺ The argument VCS$K_DECW_DISPLAY is passed to your application if the user specified the /INTERFACE=DECWINDOWS qualifier and optionally the /DISPLAY=*node::server.screen* qualifier on the VCS ACCESS command line. If your application does not provide a DECwindows interface, you should inform the user of that fact and you may also choose to terminate the application at this point.

❻ The information that you retrieved and saved in the previous steps can now be processed at this point. This is also a good time to set up an exit handler or other types of exception and error handling logic. If your application provides a DECwindows interface, you should also initialize the DECwindows Toolkit variables and environment at this point.

Routine VCS$USER_MAINLOOP.

❼ The callback routine VCS$USER_MAINLOOP is called if the user specified the /INTERFACE=DECWINDOWS qualifier and optionally the /DISPLAY=*node::server.screen* qualifier on the VCS ACCESS command line. If your application does not provide a DECwindows interface, you do not have to provide this callback routine.

❽ This loop passes through the entire argument block to retrieve the address of the DECwindows work procedure, which is provided by the Access Interface, to be registered.

❾ You may use either the DECwindows ADD WORK PROC or APPLICATION ADD WORK PROC intrinsic routine to register the work procedure. If you are planning to implement the application context in your logic, you must also pass it to the Access Interface at this point. You must specify the application context value as the *client data* parameter of the APPLICATION ADD WORK PROC intrinsic routine.

❿ Finally, enter the DECwindows MAIN LOOP intrinsic routine which is an infinite loop that processes and dispatches DECwindows events.

Routine VCS$USER_SHUTDOWN.

⓫ The callback routine VCS$USER_SHUTDOWN is called when the Access Interface is terminated, either by a user program request or by the shutting down of the VCS system initiated by the SHUTDOWN command. This callback routine is required.

⓬ This is the place to clean up after yourself, for example, canceling the exit handler and closing input and output device channels.

Routine VCS$USER_RECONFIGURE.

⓭ The callback routine VCS$USER_RECONFIGURE is called when a VCS reconfiguration event occurs. Its argument block is identical in format to that of VCS$USER_INITIALIZE. This callback routine is optional.

⓮ This loop passes through the entire argument block to retrieve the list of all nodes in the new VCS configuration.

⓯ The list of systems in the new configuration can be used to compare with that given earlier in the VCS$USER_INITIALIZE callback routine's argument block to determine if any specified system is no longer in the VCS configuration.

Routine VCS$USER_SCAN_EVENT.

⓰ The callback routine VCS$USER_SCAN_EVENT is called when an event that was specified in the configuration file as requiring operator notification is detected on a console. This callback routine is optional.

⓱ This loop passes through the entire argument block to retrieve all information about the detected event.

⓲ This is where you can do the interesting things with your application. For example, the information about a scan event can be used to determine how and where to it should be disseminated. It is a good practice to devise your code such that the program flow control is not blocked at this location for too long. For example, it may not be a good idea to block the program at this point to wait for an input from the user, which may never come. If the program flow control is blocked for too long, your application may eventually get overrun by later events from the Data Scanner.

Routine VCS$USER_VCS_EVENT.

⓳ The callback routine VCS$USER_VCS_EVENT is called when an event occurs that alters the accessibility of nodes serviced by the VCS system, or when a user of VCS enters or exits an interactive interface. This callback routine is optional.

⓴ This loop passes through the entire argument block to retrieve all information about the detected event.

㉑ The information about the VCS environment, for example, the users using the interactive interfaces, can be processed at this point.

9.3 Application Design Considerations

In addition to the above suggestions, a number of other strategies, described in this section, ought to be considered when developing an Access Interface application.

9.3.1 Event Management

When an event is detected, it is stored in a ring buffer, which is of fixed size, before being passed on to the user application through the Access Interface. Over time, old events get overwritten by new events. Therefore, the life expectancy of an event is not infinite and should be treated as such. The Access Interface will not deliver the next event until you have returned program control from a callback routine. Thus, if your application spends a long time processing an event, there is a possibility it will miss some events.

To overcome this, you may want to consider the VMS mailbox facility for delivery of the event information to a detached or subprocess. This detached or subprocess could then take as much time as needed to process the event.

9.3.2 Testing Environment

You can test the application with either a live system console connection or a simulated one. Either type of testing requires that you have privileged access to the VCS host system and can install your application software there.

Testing with a live system console connection also requires that you have access to that system and can generate the necessary console messages from there, which may interfere with the normal activities of the operations staff. Testing with live system console connection should be planned for during the later stage of your application development, at which time the software is more robust and behaves reasonably well.

On the other hand, generating events on a simulated console connection is easy and by far is the most economical and efficient means of testing the software at the early stage of development. At this stage, you need a facility which can generate console messages reliably and on demand. This can be done by connecting two terminal ports together, using a regular null modem cable.

One of these ports must be on either the VCS host system or a DECserver terminal server which is accessible from the VCS host system. If the terminal port resides on a terminal server, you should follow the instructions in the *VAXcluster Console System Installation Guide* to set up that port for host initiated connection and to create and map an LTA device on the VCS host system to that same port. The device name of this terminal port is the one you will enter into the VCS configuration database when you create a fictitious node for testing.

The second port can be a time-sharing terminal port on any VAX/VMS system. It can also be another time-sharing terminal port on the VCS host system or the same terminal server as the first one. Ideally, this terminal port should be one on a VAX/VMS system that you may log in without requiring any special privileges. This port will be the simulated console port for your testing.

To test the null modem cable connection, use the DCL command SET HOST/DTE from a logged in terminal on the VCS host system, specifying the device name of the first terminal port of the two described above. If the connection is operational, you can enter and receive data on the logged in terminal display. Enter Ctrl/ \ to escape back to your currently logged in terminal session.

After the terminal port connection has been established and tested, you may proceed to add a fictitious node to the VCS configuration database. Use the VCS RECONFIGURE command to activate the newly entered information. From that point on, ASCII data from the regular time-sharing terminal port (the simulated console port) is logged in the VCS log file and displayed as the console data from the fictitious system. Generating an event can now be done through a variety of means.

If the simulated console port is a regular time-sharing terminal port on a remote system, you should use the Connect Interface to connect and log in to that VAX/VMS system. At DCL level on that system, you should be able to use various DCL commands and procedures to generate simulated console messages repeatedly and as needed.

If the simulated console port is a DECserver port, you may use the Connect Interface first to connect to a VAX/VMS system and then proceed as described above. However, you may also use the Monitor Interface OUTPUT command to generate simulated console messages by sending the appropriate text strings to the terminal server at the "Local>" prompt. These text strings are echoed by the terminal server as illegal commands.

9.3.3 Logic Verification

You may use the Access Interface Default Application, described in Chapter 5, to verify your application logic. The Access Interface Default Application provides a simple display of all events detected by VCS, complete with time stamp and other information associated with an event. This information could be used to cross-check the information that your application believes it has or has not received.

9.4 Linking and Executing User Code

When you start the Access Interface process, it examines the command line to see whether you included an /APPLICATION qualifier to point to a shareable image file. If so, it looks for a shareable image and determines if the required callback routines are present. If they are present, the shareable image is then dynamically activated into the P0 address space of the process and execution begins. If the required routines are not present, the process terminates.

To create the shareable image, you must link the application with the /SHARE qualifier. You must also use an option file to define the routine names as universal symbols. For example, to link the shareable image VCS$APPLICATION.EXE:

```
$ LINK/SHARE/NODEBUG/NOTRACE VCS$APPLICATION.OBJ,SYS$INPUT/OPTION
UNIVERSAL=VCS$USER_INITIALIZE, -
VCS$USER_MAINLOOP, -
VCS$USER_SHUTDOWN, -
VCS$USER_VCS_EVENT, -
VCS$USER_SCAN_EVENT, -
VCS$USER_RECONFIGURE
^Z
```

The VCS ACCESS command activates the image VCS$IFEX.EXE, which is an installed image with privileges. Therefore, the user application must be installed with the same privileges as VCS$IFEX.EXE by executing the following commands:

```
$ COPY VCS$APPLICATION.EXE SYS$SHARE
$ INSTALL CREATE SYS$SHARE:VCS$APPLICATION.EXE -
_$ /PRIVILEGES=(CMEXEC,SYSLCK,READALL,PRMGBL)
```

Note that when the application is installed with the INSTALL program, a problem can arise from external or global data declarations. When this problem occurs, the following messages are displayed:

```
%VCS-W-INVAPPL, Error with user application file application
-VCS-E-NOENTRY, Entry VCS$USER_INITIALIZE is required but not found
%SYSTEM-F-PRIVINSTALL, shareable images must be installed to run privileged image
```

The *VAX/VMS Linker Reference Manual* contains a discussion of shareable images that addresses the problem. The WRT and NOSHR attributes can be assigned to psects to ensure that each incarnation of the application has its own internal data fields. Most programming languages provide a second method for accomplishing the same thing, such as the *noshare* storage-class declaration modifier of VAX C.

After you have debugged the application and installed it in SYS$SHARE, you can define a logical name to point to the file name as the argument to the /APPLICATION qualifier in the VCS ACCESS command. This logical name must be a system-wide, executive-mode logical name, or the Access Interface cannot locate the shareable image. Define such a logical name and activate the application as follows:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE logical-name file-name
$ VCS ACCESS node-name[,...] /APPLICATION=logical-name
```

For example:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE VCS$APPLICATION VCS$APPLICATION.EXE
$ VCS ACCESS VCS$ALL_NODES /APPLICATION=VCS$APPLICATION
```

As a result of the above commands, the Access Interface looks in SYS$SHARE for the image file VCS$APPLICATION.EXE.

9.5 Debugging User Code

To enable you to access the application code through the VAX/VMS Debugger, the product software kit contains a debugging version of the module VCS$IFEX.EXE. This file is called VCS$IFEX.DBG. You can access VCS$IFEX.DBG by creating a logical name for VCS$IFEX that translates to it.

After the user shareable image is dynamically activated into the P0 address space of the process, the routine VCS$USER_BREAKPOINT is called. This is an empty routine which is called so that you can set a break point to halt the program execution. When this break point is reached, you can load the application's symbol table by using the Debugger SET IMAGE command. Once the symbols are available, then you can follow normal debugging procedures.

After defining a system-wide executive-mode logical name to point to the user shareable image file as shown in Section 9.4, perform the following command sequence to set a break point in the user application code:

- Use the debugging version of VCS$IFEX, VCS$IFEX.DBG:

  ```
  $ DEFINE VCS$IFEX SYS$SYSTEM:VCS$IFEX.DBG
  ```

- Activate the Access Interface:

  ```
  $ VCS ACCESS node-name[,...] /APPLICATION=logical-name
  ```

- Set the initial break point and start the process:

```
DBG>SET BREAK VCS$USER_BREAKPOINT
DBG>GO

break at routine VCS$IFEX_MAIN\VCS$USER_BREAKPOINT
%DEBUG-W-UNAOPNSRC, unable to open source file
     DUA0:[VCS]VCS$IFEX_MAIN.C;1
-RMS-F-DEV, error in device name or inappropriate
      device type for operation
DBG>
```

  The warning message is generated because the source for VCS$IFEX_MAIN.C cannot be opened. This warning may be ignored.

- Load the application symbol table:

```
DBG>SET IMAGE logical-name
DBG>SET MODULE /ALL
DBG>SET BREAK VCS$USER_INITIALIZE
DBG>SET BREAK VCS$USER_MAINLOOP
DBG>SET BREAK VCS$USER_VCS_EVENT
DBG>SET BREAK VCS$USER_SCAN_EVENT
DGB>SET BREAK VCS$USER_RECONFIGURE
DBG>SET BREAK VCS$USER_SHUTDOWN
DBG>GO
```

  The above commands can be placed in a debugger initialization file for easier use.

For example:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE VCS$APPLICATION VCS$APPLICATION.EXE
$ DEFINE VCS$IFEX SYS$SYSTEM:VCS$IFEX.DBG
$ VCS ACCESS VCS$ALL_NODES /APPLICATION=VCS$APPLICATION
       VAX DEBUG Version  V5.3-00

%DEBUG-I-INITIAL, language is C, module set to VCS$IODL_CTRL
DBG>SET BREAK VCS$USER_BREAKPOINT
DBG>GO

break at routine VCS$IFEX_MAIN\VCS$USER_BREAKPOINT
%DEBUG-W-UNAOPNSRC, unable to open source file
     DUA0:[VCS]VCS$IFEX_MAIN.C;1
-RMS-F-DEV, error in device name or inappropriate
      device type for operation
DBG>SET IMAGE VCS$APPLICATION
DBG>SET MODULE /ALL
DBG>SET BREAK VCS$USER_INITIALIZE
DBG>SET BREAK VCS$USER_MAINLOOP
DBG>SET BREAK VCS$USER_VCS_EVENT
DBG>SET BREAK VCS$USER_SCAN_EVENT
DGB>SET BREAK VCS$USER_RECONFIGURE
DBG>SET BREAK VCS$USER_SHUTDOWN
DBG>GO
```

9.6 Customizing the DCL Command Syntax

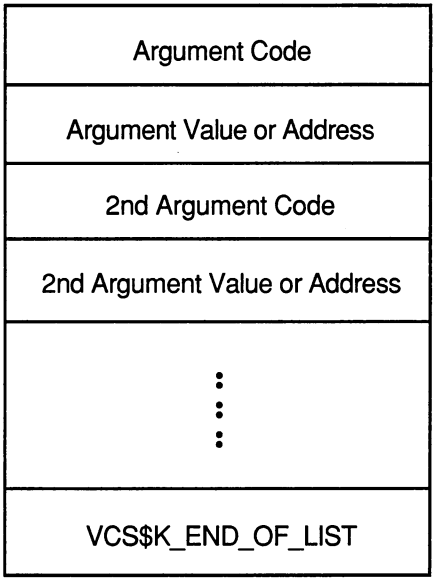
The definition of the DCL command syntax is contained in the file VCS$EXAMPLES:VCS$DCL.CLD.

You can modify this file to add new parameters and qualifiers to the VCS ACCESS command syntax. You must process these qualifiers in the VCS$USER_INITIALIZE routine. Refer to the *VAX/VMS Command Definition Utility Reference Manual* for additional information on how to add the new qualifiers and parameters as well as how to retrieve their values.

9.7 Callback Routine Reference

This section describes the Access Interface calling conventions for the callback routines. Each callback routine, except the routine VCS$USER_SHUTDOWN, receives a single argument which is the address of an item list. The list is a series of longword pairs. The first longword contains an argument code and the second longword contains the argument value or its address. The end of the item list is signified by the argument code VCS$K_END_OF_LIST. Figure 9–1 contains the format of the item list argument block. The order of the arguments and the number of arguments may change. It is important that your code test each argument code.

Figure 9–1 VCS Access Interface Argument Block



MR–1774–RA

The argument codes are defined in a common source file to be included in the user code. Table 9–2 in Section 9.1 lists the common source files for various programming languages which have been placed in SYS$LIBRARY as part of the product installation.

The remainder of this section contains reference information about the Access Interface callback routines. The callback routines are listed in alphabetical order.

VCS$USER_INITIALIZE

This routine is the first one called within the user application. You must open any necessary files, interpret the command line arguments, and set up any desired exit handlers and data structures within this routine.

Format

VCS$USER_INITIALIZE *itmlst*

Returns

No return value; the routine should fully process any error condition.

Arguments

itmlst
VMS Usage: **item_list_pair**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference or descriptor**

VCS$USER_INITIALIZE

The arguments in Table 9–3 are present in the **itmlst**:

Table 9–3 VCS$USER_INITIALIZE Argument List

Argument Code	Argument Value	Description
VCS$K_SYSTEM_COUNT	Unsigned longword	The number of systems specified in the command line.
VCS$K_SYSTEM_LIST	Address of an item list pair	The list of systems specified in the command line. The list is a series of longword pairs. The first longword contains the argument code VCS$K_SYSTEM_NAME, and the second longword contains the address of a descriptor pointing to a system name. The end of the item list is signified by the argument code VCS$K_END_OF_LIST.
VCS$K_DECW_DISPLAY	Address of a descriptor	The descriptor points to a zero length character string or one that contains the name of the DECwindows display server in the format of *node::server.screen*. Either type of this character string is to be passed as an argument to the DECwindows OPEN DISPLAY intrinsic routine.
VCS$K_END_OF_LIST	None	Terminates item list. This is the last longword in the attribute list. It does not require an associated longword.

Description

VCS$USER_INITIALIZE enables the user application to process the data entered on the command line that invokes the Access Interface. The user application can initialize data storage, input/output devices, and so on. An optional exit handler can also be set up in this routine by using the system call $DCLEXH to provide for cleanup if the program is terminated abnormally.

If your application provides a DECwindows interface, you should also initialize the DECwindows Toolkit variables and environment, and open the display using the information specified by the argument VCS$K_DECW_DISPLAY.

The argument VCS$K_DECW_DISPLAY is passed to your application only if the user specified the /INTERFACE=DECWINDOWS and the optional /DISPLAY=*node::server.screen* qualifiers on the VCS ACCESS command line. If your application does not provide a DECwindows interface, you should inform the user of that fact and you may also choose to terminate the application at this point.

After this routine is executed, if your application does not provide a DECwindows interface, the Access Interface will not call another routine until a scan event has been detected by the Data Scanner, a VCS event has occurred within VCS, or a VCS reconfiguration has occurred. Otherwise, the routine VCS$USER_MAINLOOP is called after the Access Interface completes its initialization.

VCS$USER_MAINLOOP

This routine is called during the initialization process of the user DECwindows application. It provides a mechanism for interfacing to DECwindows.

Format

VCS$USER_MAINLOOP *itmlst*

Returns

No return value; the routine should fully process any error condition.

Arguments

itmlst
VMS Usage: **item_list_pair**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference or descriptor**
The arguments in Table 9–4 are present in the **itmlst**:

Table 9–4 VCS$USER_MAINLOOP Argument List

Argument Code	Argument Value	Description
VCS$K_WORK_ADDRESS	Procedure entry mask	The address of a DECwindows work procedure belonging to the Access Interface to be registered.
VCS$K_END_OF_LIST	None	Terminates item list. This is the last longword in the attribute list. It does not require an associated longword.

Description

VCS$USER_MAINLOOP enables the user application to interface to DECwindows. This routine is called when the Access Interface is ready to process events.

The argument value of VCS$K_WORK_ADDRESS is the address of a DECwindows work procedure belonging to the Access Interface that is to be registered using the DECwindows ADD WORK PROC or APPLICATION ADD WORK PROC intrinsic routine. You must pass the address of this work procedure as an argument to the intrinsic routine.

The address of this work procedure can be used with either the default DECwindows application context or a user created application context. If your application uses the DECwindows INITIALIZE intrinsic routine, you can use ADD WORK PROC to register the work procedure. In this case, the ADD WORK PROC parameter *client data* must be set to zero.

If your application uses the DECwindows CREATE APPLICATION CONTEXT intrinsic routine to create an application context, you must use APPLICATION ADD WORK PROC to register the work procedure. Since the application context is to be used internally by the Access Interface, it must be passed as the APPLICATION ADD WORK PROC parameter *client data*.

For more information on the application context, refer to Section "Initializing the XUI Toolkit" of the *VMS DECwindows Guide to Application Programming*.

After registering the work procedure, the final step is to enter the DECwindows MAIN LOOP or APPLICATION MAIN LOOP intrinsic routine. Either routine is not expected to return. The program control will later be returned to your application through one of the DECwindows or other VCS callback routines.

The VCS$USER_MAINLOOP routine is not expected to return.

VCS$USER_RECONFIGURE

This routine is called when a VCS reconfiguration event occurs. This event generally does not require operator action; it is considered advisory.

Format

VCS$USER_RECONFIGURE *itmlst*

Returns

No return value; the routine should fully process any error condition.

Arguments

itmlst
VMS Usage: **item_list_pair**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference or descriptor**

The arguments in Table 9–5 are present in the **itmlst**:

Table 9–5 VCS$USER_RECONFIGURE Argument List

Argument Code	Argument Value	Description
VCS$K_SYSTEM_COUNT	Unsigned longword	The number of systems in the new configuration.
VCS$K_SYSTEM_LIST	Address of an item list pair	The list of systems in the new configuration. The list is a series of longword pairs. The first longword contains the argument code VCS$K_SYSTEM_NAME, and the second longword contains the address of a descriptor pointing to a system name. The end of the item list is signified by the argument code VCS$K_END_OF_LIST.
VCS$K_END_OF_LIST	None	Terminates item list. This is the last longword in the attribute list. It does not require an associated longword.

Description

VCS$USER_RECONFIGURE enables the user application to process the new VCS configuration information when the command RECONFIGURE is issued.

The system list does not include any system names that were not included in the user's original command line. The list specifies which systems are still active.

VCS$USER_SCAN_EVENT

This routine is called when an event that is specified in the configuration file occurs on a node. It enhances the current scan event processing by enabling the user software to examine all events as they are detected by VCS.

Format

VCS$USER_SCAN_EVENT *itmlst*

Returns

No return value; the routine should fully process any error condition.

Arguments

itmlst
VMS Usage: **item_list_pair**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference or descriptor**
The arguments in Table 9–6 are present in the **itmlst**:

Table 9–6 VCS$USER_SCAN_EVENT Argument List

Argument Code	Argument Value	Description
VCS$K_SYSTEM_NAME	Address of a descriptor	The descriptor points to a character string containing the name of the node on which the event occurred.
VCS$K_EVENT_NAME	Address of a descriptor	The descriptor points to a character string containing the event name in full as defined in the configuration file.

(continued on next page)

Table 9–6 (Cont.) VCS$USER_SCAN_EVENT Argument List

Argument Code	Argument Value	Description
VCS$K_EVENT_ABBREVIATION	Address of a descriptor	The descriptor points to a character string containing the event name abbreviation up to 6 characters in length.
VCS$K_SCAN_TEXT	Address of a descriptor	The descriptor points to a character string containing the line of console data. This line of text contains a substring that was detected by the VCS Data Scanner.
VCS$K_SCAN_MATCH	Unsigned longword	The left half contains the offset of the substring within the text string VCS$K_SCAN_TEXT. The right half contains the length of the substring.
VCS$K_PRIORITY	Unsigned longword	The priority assigned to the scan event.
VCS$K_TIME	Address of an unsigned quadword	The time the event occurred. This is a time stamp of the console log record containing the scan event text.
VCS$K_CONTEXT_START	Address of an unsigned quadword	The time stamp of the first context record in the console log file.
VCS$K_CONTEXT_COUNT	Unsigned longword	The number of console text lines following the first context line that contains relevant information.
VCS$K_INFORMATION	Address of a descriptor	The descriptor points to a character string containing the optional user-defined information field from the scan event record.
VCS$K_END_OF_LIST	None	Terminates item list. This is the last longword in the attribute list. It does not require an associated longword.

VCS$USER_SCAN_EVENT

Description

VCS$USER_SCAN_EVENT provides you with the necessary information for deciding if additional operator notification or action is required as the result of an event that has occurred on a system node.

For example, the user routine can display the information on a terminal or try to act upon it through mail or submission of a batch process. You can choose to disregard some messages or record them in a data file for later processing. The *information* field of the event record in the configuration file can be modified to provide additional information to the application.

The time of the event can be used to locate the console log file record containing the scan event text. The *context start* is the time stamp of the first log record containing related information. The *context count* is the number of log records surrounding the event record which may be of interest. The context fields are generated by the Data Scanner by using the information in the event records while examining console data.

A set of local event records can be established to capture classes of messages to be collected in a data file that will processed later for system analysis. Local applications running on the serviced systems can generate events by using the system service call $SNDOPR.

VCS$USER_SHUTDOWN

This routine is called when the Access Interface is terminated, either by a user program request or by the shutting down of the VCS system initiated by the SHUTDOWN command.

Format

VCS$USER_SHUTDOWN

Returns

No return value; the routine should fully process any error condition.

Description

You must terminate processing as appropriate for the application. Files should be closed, terminals should be reset, and all appropriate user messages should be sent.

VCS$USER_VCS_EVENT

This routine is called when an event occurs that alters the accessibility of nodes serviced by the VCS system, or a user of VCS enters or exits. These events generally do not require operator action; they are considered advisory.

Format

VCS$USER_VCS_EVENT *itmlst*

Returns

No return value; the routine should fully process any error condition.

Arguments

itmlst
VMS Usage: **item_list_pair**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference or descriptor**
The arguments in Table 9–7 are present in the **itmlst**:

Table 9–7 VCS$USER_VCS_EVENT Argument List

Argument Code	Argument Value	Description
VCS$K_TIME	Address of an unsigned quadword	The time the event occurred.
VCS$K_SYSTEM_NAME	Address of a descriptor	The descriptor points to a character string containing the name of the system that is affected by the event.

(continued on next page)

Table 9–7 (Cont.) VCS$USER_VCS_EVENT Argument List

Argument Code	Argument Value	Description
VCS$K_USER_PID	Unsigned longword	The user process ID of the process that caused the event.
VCS$K_EVENT_CODE	Unsigned longword	A code indicating the type of VCS event. Table 9–8 contains the code types.
VCS$K_EVENT_TEXT	Address of a descriptor	The descriptor points to an ASCII message describing the event which has taken place. It is suitable for printing or writing to a file.
VCS$K_END_OF_LIST	None	Terminates item list. This is the last longword in the attribute list. It does not require an associated longword.

Table 9–8 VCS Event Codes

Advisory Code	Description
VCS$K_DISABLE	Service to a system has been disabled.
VCS$K_ENABLE	Service to a system has been enabled.
VCS$K_ERROR	A VCS error has been detected.
VCS$K_EVENT	An unspecified VCS event has been detected.
VCS$K_LOCK	A system has been locked.
VCS$K_RECONFIG	A VCS reconfiguration has occurred.
VCS$K_TERMINATION	VCS has been shutdown.
VCS$K_UNLOCK	A system has been unlocked.
VCS$K_USRSIGNON	New user of VCS.
VCS$K_USRSIGNOFF	User of VCS has exited.

VCS$USER_VCS_EVENT

Description

VCS$USER_VCS_EVENT provides you with an update on the status of the VCS environment.

The argument VCS$K_USER_PID will only be delivered in conjunction with the following event codes:

- VCS$K_LOCK
- VCS$K_UNLOCK
- VCS$K_USRSIGNON
- VCS$K_USRSIGNOFF

10

Event Notification System

The Event Notification System (ENS) is a VCS Access Interface application. The ENS software, upon receiving an event from VCS, can selectively perform any of the following:

- Broadcast to any specified VCS terminal or user.
- Send VAXmail to an account on any system.
- Speak a message using a DECtalk system.
- Create a notification window on any DECwindows screen.

ENS automatically sends notification reports using any of the preceding mechanisms to call attention to the event.

In addition to using these pre-defined routines, ENS also enables you to add notification procedures to fit your operational needs. An example of such a procedure might be a beeper routine which automatically dials a beeper service with a special code. Another possible example might be to start up a repair procedure or program on a troubled system that alleviates the problem at that system.

This chapter includes the following topics:

- Getting Started
- ENS Action Routines
- Adding Your Own Action Routines
- ENS Editor Commands

10.1 Getting Started

ENS runs in a passive mode and waits for an event to occur before performing any work. Upon receiving an event, ENS processes the event, then hibernates until another event occurs. ENS maintains the ENS data file that defines the events to watch for and the corresponding notification actions to execute.

10.1.1 Starting ENS

The ENS software begins running at VCS startup. To verify that ENS is running, issue the DCL command SHOW SYSTEM and watch for a process called VCS$ENS. If the process has been stopped, you may restart ENS by issuing the following command:

```
$  @SYS$STARTUP:VCS$STARTUP VCS$ENS Return
```

10.1.2 Using the ENS Editor

As previously noted, the ENS data file defines what events to watch for and what notification actions to perform when a specific event occurs. To edit the ENS data file, use the ENS Editor. To run the ENS Editor, enter the following DCL command:

```
$  RUN SYS$SYSTEM:VCS$ENS_EDITOR Return
```

The following prompt appears on the screen:

```
ENS EDIT>
```

Within the ENS data file are two types of entries: *events* and *actions*.

- *Event entries* determine which events from VCS that ENS should act upon. A single event can trigger multiple actions. For example, the ENS data file can be constructed to perform several ENS actions, such as sending VAXmail or speaking a DECtalk message, when a single event occurs. Conversely, multiple events can be defined to perform the same ENS action.
- *Action entries* specify the program names and parameters. ENS supplies five pre-defined action routine definitions in the ENS data file. These pre-defined routines are discussed in Section 10.2. You can also define your own action routines (see Section 10.3).

For full information about the commands for defining events and actions, see Section 10.4.

The following example shows the command syntax you could use to add an event named HALT.

```
ENS EDIT> ADD EVENT HALT /ACT1 = (NAME = MAIL, DEST = KAPPA::SMITH) - Return
_ENS EDIT> /SYSTEM = * Return
```

This command directs ENS to watch for the event named HALT. When the HALT event occurs, ENS notifies user SMITH by way of mail. This example introduces three new elements:

- **/ACT1** specifies the name of the action and the number of actions being defined. In this example, the first action is being defined. A maximum of eight actions can be defined to take place when an event occurs. Note that no validity checking is done. (SMITH may not have an account on node KAPPA.)
- **NAME** specifies an action-name. The action-name is an entry containing information about which program to run and which parameters are to be passed. In this example, MAIL has been selected.
- **DEST** specifies a value that is passed to the action routine. In this example, a request is being made that user SMITH on node KAPPA be sent some mail.

Both the NAME and DEST keywords are required when you add a new event to the ENS data file.

Systems and events can also be paired through the use of the /SYSTEM qualifier, as shown in the following example:

```
ENS EDIT> ADD EVENT HALT /ACT1 = (NAME = MAIL, DEST = KAPPA::SMITH) - Return
_ENS EDIT> /SYSTEM = (ALPHA, BETA, GAMMA) Return
```

In this example, SMITH receives mail only if event HALT occurs on nodes ALPHA, BETA, or GAMMA. If event HALT occurs on all three nodes, then SMITH receives three mail messages.

More than one action can be performed for the same event. In the following example, a second action is being added to the event HALT:

```
ENS EDIT> ADD EVENT HALT /ACT1 = (NAME = MAIL, DEST = KAPPA::SMITH) - Return
_ENS EDIT> /ACT2 = (NAME = DECTALK, DEST = TXA0:) - Return
_ENS EDIT> /SYSTEM = (ALPHA, BETA, GAMMA) Return
```

To remove an action from an event definition, enter the following command:

```
ENS EDIT> MODIFY EVENT HALT /ACT2 = REMOVE Return
```

ENS must be notified after you make changes to the ENS data file. To notify ENS, issue the DCL command VCS RECONFIGURE. Any additions, modifications, or deletions to the ENS data file are now activated.

10.1.3 Testing ENS

To test ENS, use the VCS CONNECT command to log onto a system console. Next, enter the command syntax to trigger the event you have defined. Additional information on how to simulate events can be found in Chapter 9.

10.1.4 Log Files

For each notification, ENS creates a log file which is deleted after *successful* job completion. The ENS log files are located in the VCS$ENS_LOG directory and have the format VCS$ENS_ 0000.LOG for command files. The 0000 is an event number (which is incremented for each event triggering a command file) and *process-name*.LOG for a program sent from a mailbox.

If VCS$DEBUG is set to TRUE, the log files are not deleted.

10.1.5 Logical Names

The following is a list of logical names used by ENS:

- VCS$ENS - Working directory for access interface
- VCS$ENS_DATA_FILE - Data file
- VCS$ENS_LOG - Directory containing log files of any notification procedure that terminates with a failure code.

To review these and other present miscellaneous names, enter the following command:

```
$ SHOW LOGICAL VCS$ENS* Return
```

10.2 ENS Action Routines

The five pre-defined ENS action routines are listed as follows:

- Brdcst
- DECtalk
- Mail
- Window
- Window1

These ENS action routines are described in the following subsections.

10.2.1 Brdcst

The Brdcst action routine passes information about an event to a system terminal or user in the form of a five-line message as shown in the following example:

```
System   : GAMMA
Event    : HALT
Priority : 90
Time     : 27-DEC_1989 11:57:44.95
Text     : System Halted
```

The user or terminal is valid only for users who are logged onto the VCS host system. This information cannot be displayed on a terminal connected to another system.

When adding an event you want to have broadcasted, use Brdcst as the value for the NAME keyword in the /ACTn qualifier. Refer to the ADD EVENT command in Section 10.4 for more information. When entering the value for the DEST keyword, the **#u** sign or the **#t** sign must be appended to the value as a type identifier. The **#u** sign signifies a user and the **#t** sign signifies a terminal.

In the following example, the event HALT is added to the ENS data file. Brdcst sends a formatted message whenever node ALPHA is halted.

```
ENS EDIT>  ADD EVENT HALT /ACT1 = (NAME=BRDCST, DEST=SMITH#U) - Return
_ENS  EDIT> /SYSTEM = ALPHA Return
```

In the following example, the event CICRA is added to the ENS data file. Brdcst sends a formatted message to any user logged on the VCS host system if the CI port has crashed on any serviced node.

```
ENS EDIT>  ADD EVENT CICRA /ACT1 = (NAME=BRDCST, DEST=ALL#U) - Return
_ENS EDIT>  /SYSTEM = * Return
```

10.2.2 Mail

The Mail action routine sends a message to those destinations specified in the event definitions in the ENS data file. The Mail action routine is a DCL command procedure called VCS$ENS_VAXMAIL.COM. You can find this command procedure in the directory VCS$ENS.

The contents of the message can be changed to suit your needs by editing VCS$ENS_VAXMAIL.COM. The following is an example of such a message.

```
From:    GAMMA::BIGELOW
To:      ALPHA::MANAGER
CC:
Subj:    VCS Event Notification

VCS Event Notification System

  System : BETA
   Event : HALT
Priority : 95
    Time : 18-DEC-1989 14:15:46.05
    Text : System Halted
```

When adding or modifying an event entry in the ENS data file, use Mail as the value for the NAME keyword and mail addresses for the value of the DEST keyword. Multiple addresses may be used if separated by commas. If entering a list, enclose the list in quotation marks.

The following example shows how to modify the HALT event.

```
ENS EDIT> MODIFY EVENT HALT /SYSTEM = ALPHA - Return
_ENS EDIT> /ACT2 = (NAME=MAIL, - Return
_ENS EDIT> DEST="BETA::FIELD_SERVICE, GAMMA::CARLETON") Return
```

In this example, the ENS event HALT is modified. The action routine MAIL sends a message to users FIELD_SERVICE and CARLETON if VCS detects a halt event on node ALPHA.

In the following example, the event CICRA is modified. Mail sends a message to user STEVENS if the CI port has crashed on any node.

```
ENS EDIT> MODIFY EVENT CICRA /ACT2 = (NAME=MAIL, DEST=STEVENS) - Return
_ENS EDIT> /SYSTEM = * Return
```

10.2.3 DECtalk

The DECtalk action routine sends a message to a DECtalk unit for any event that has occurred. Since a nodename may not be easily pronounced, DECtalk spells out the nodename. The DECtalk unit can be local to the VCS host system or reside on a host initiated connection anywhere on the LAN. DECtalk speaks the line of text that triggers the event.

Specify DECtalk as the value of the NAME keyword and the terminal port as the value for the DEST keyword of the /ACTn qualifier.

```
ENS EDIT> ADD EVENT BOOT /SYSTEM = ALPHA - Return
_ENS EDIT> /ACT3 = (NAME=DECTALK, DEST=TXA1:) Return
```

In this example, the ENS event BOOT is added. DECtalk outputs a message to the unit located on terminal line TXA1: if the BOOT event occurs on any node.

10.2.4 Window

The Window action routine is a program that enables an ENS scroll window to be displayed on any workstation on your network. This scroll window displays error severity by color. The error or information messages disappear once they have been acknowledged by the workstation user (by double clicking MB1 on the event).

If the window is reduced to an icon, the icon changes color reflecting the highest event priority.

To activate the Window routine, enter an event entry into the ENS data file and specify Window as the value for the NAME keyword of the /ACTn qualifier. The output displays can be specified as the values of the DEST keyword.

Workstations that are to receive messages from ENS need to have security parameters set correctly to allow ENS to create a window. Set up workstation security using the Customize menu of the Session Manager.

Figure 10–1 shows an example of an ENS event window.

Figure 10–1 ENS Window

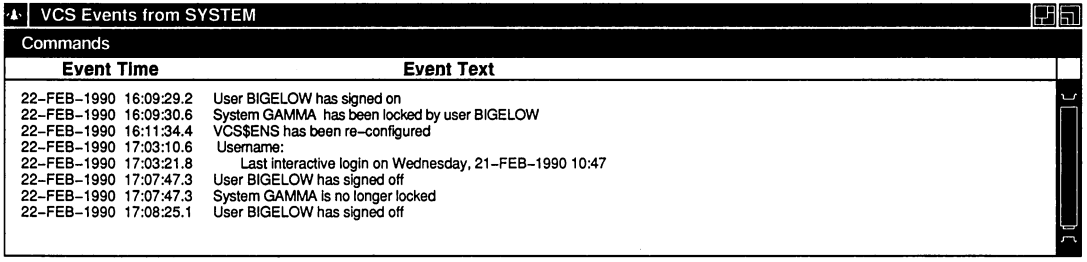
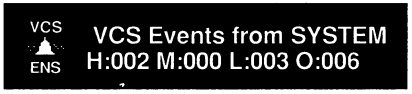

VCS Events from SYSTEM
Commands
Event Time
Event Text
22-FEB-1990 16:09:29.2 User BIGELOW has signed on
22-FEB-1990 16:09:30.6 System GAMMA has been locked by user BIGELOW
22-FEB-1990 16:11:34.4 VCS$ENS has been re-configured
22-FEB-1990 17:03:10.6 Username:
22-FEB-1990 17:03:21.8 Last interactive login on Wednesday, 21-FEB-1990 10:47
22-FEB-1990 17:07:47.3 User BIGELOW has signed off
22-FEB-1990 17:07:47.3 System GAMMA is no longer locked
22-FEB-1990 17:08:25.1 User BIGELOW has signed off

Figure 10–2 shows the ENS Window action routine icon. The icon shows how many events of each priority level that ENS has received.

Figure 10–2 ENS Window Action Routine Icon



The following command directs ENS to notify workstations ALPHA and BETA when a mount request occurs on any serviced node.

```
ENS EDIT> ADD EVENT MOUNT - Return
_ENS EDIT> /ACT5 = (NAME=WINDOW, DEST=(ALPHA,BETA) ) - Return
_ENS EDIT> /SYSTEM = * Return
```

10.2.4.1 Window Customization You can customize your ENS window by modifying the contents of the file DECW$XDEFAULTS.DAT in your DECW$USER_DEFAULTS directory (SYS$LOGIN) on the system where you want the window to appear (the DEST parameter in the event entry).

You must log out of your current session and then log back in to activate your ENS window customization settings. The customization settings are not applied when VCS is started at system startup.

Table 10–1 describes the DECwindows resources, default values, and value ranges available for you to customize the window routine.

Table 10–1 ENS Window Routine Customization Parameters

DECwindows Resource	Default Value	Value Range
VCS$ENS.EnsForeground	White	Any valid DECwindows named color
VCS$ENS.EnsBackground	Black	Any valid DECwindows named color
VCS$ENS.VCS100	Blue	Any valid DECwindows named color
VCS$ENS.VCS99	White	Any valid DECwindows named color
VCS$ENS.HighPriority	Red	Any valid DECwindows named color
VCS$ENS.MediumPriority	Yellow	Any valid DECwindows named color
VCS$ENS.LowPriority	Green	Any valid DECwindows named color
VCS$ENS.VCS01	LightGray	Any valid DECwindows named color

(continued on next page)

Table 10–1 (Cont.) ENS Window Routine Customization Parameters

DECwindows Resource	Default Value	Value Range
VCS$ENS.X	Center	"Center" or any valid Screen Pixel Value
VCS$ENS.Y	Center	"Center" or any valid Screen Pixel Value
VCS$ENS.Width	800	Any valid Screen Pixel Value
VCS$ENS.Height	200	Any Valid Screen Pixel Value
VCS$ENS.DisplayNode	Yes	Yes or No, True or False
VCS$ENS.DisplayTime	Yes	Yes or No, True or False
VCS$ENS.DisplayText	Yes	Yes or No, True or False
VCS$ENS.DisplayPriority	No	Yes or No, True or False
VCS$ENS.DisplayName	No	Yes or No, True or False
VCS$ENS.DisplayInfo	No	Yes or No, True or False
VCS$ENS.Font	Fixed	Any Font valid for your Server
VCS$ENS.Bell	25	Any Value in the range 0 to 100
VCS$ENS.Enable	True	Yes or No, True or False
VCS$ENS.MinimumPriority	0	Any valid VCS Priority
VCS$ENS.ExcludeEvents	None	Any valid event

These DECwindows resources are case sensitive. The following example, taken from the DECW$XDEFAULTS.DAT file, shows how they should be entered.

```
!
! ENS Window
!
VCS$ENS.EnsBackground:          NavyBlue
VCS$ENS.EnsForeground:          SkyBlue
VCS$ENS.VCS100:                 Blue
VCS$ENS.VCS99:                  White
VCS$ENS.HighPriority:           Red
VCS$ENS.MediumPriority:         Yellow
VCS$ENS.LowPriority:            Green
VCS$ENS.VCS01:                  LightGray
VCS$ENS.X:                      Center
VCS$ENS.Y:                      Center
VCS$ENS.Width:                  800
VCS$ENS.Height:                 200
VCS$ENS.DisplayNode:            yes
VCS$ENS.DisplayTime:            yes
VCS$ENS.DisplayText:            yes
VCS$ENS.DisplayPriority:        yes
VCS$ENS.DisplayName:            yes
VCS$ENS.DisplayInfo:            yes
VCS$ENS.Bell:   1
```

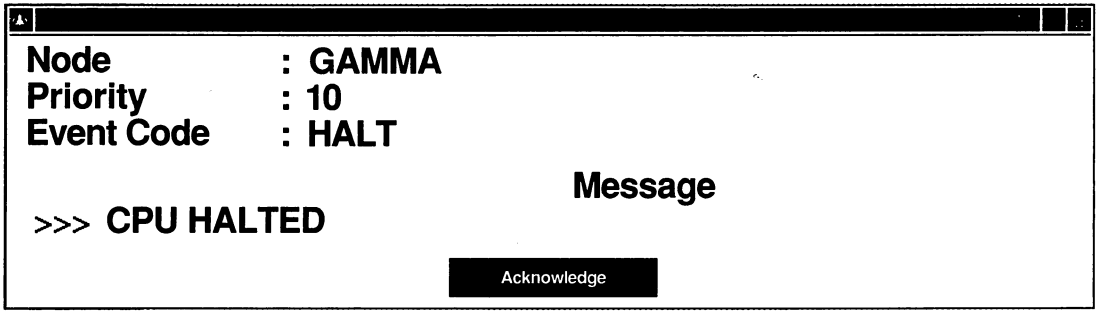
10.2.5 Window1

The Window1 action routine is a program that displays a large ENS window on any workstation on your network. Because the Window1 display is much larger than the Window display, it is, therefore, ideally suited for severe errors. The Window1 program displays one window for each event.

To close the Window1 window, click MB1 on the Acknowledge button. By default, if the message is not acknowledged within ten minutes, the window automatically closes. To modify the life of the window, you must modify parameter P5 of the Window1 action routine. Refer to Section 10.3 for more information on how to modify values.

Figure 10–3 shows an example of an ENS Window1 message.

Figure 10–3 ENS Window1



ENS creates a separate process for each event. Because each Window1 routine remains active until you acknowledge the message, the VCS host system can become overloaded.

In the following example, workstations BETA and ALPHA are notified when a CI port has crashed on any system that VCS is monitoring. The destination workstations must be customized to allow ENS to display Window1 messages on their screens.

```
ENS EDIT> ADD EVENT CICRA /ACT3 = (NAME=WINDOW1, DEST="BETA ALPHA" ) - Return
_ENS EDIT>   /SYSTEM = * Return
```

10.2.5.1 Window1 Customization

You can customize your window display by modifying the contents of the file DECW$XDEFAULTS.DAT in your DECW$USER_DEFAULTS directory (SYS$LOGIN) on the system where you want the window to appear (the DEST parameter in the event entry).

Table 10–2 describes the DECwindows resources, default values, and value ranges available for you to customize the window routine.

Table 10–2 ENS Window1 Routine Customization Parameters

DECwindows Resource	Default Value	Value Range
VCS$ENS.Window1 EnsForeground	White	Any valid DECwindows named color
VCS$ENS.Window1 EnsBackground	Black	Any valid DECwindows named color
VCS$ENS.Window1 VCS100	Blue	Any valid DECwindows named color
VCS$ENS.Window1 VCS99	White	Any valid DECwindows named color
VCS$ENS.Window1 HighPriority	Red	Any valid DECwindows named color
VCS$ENS.Window1 MediumPriority	Yellow	Any valid DECwindows named color
VCS$ENS.Window1 LowPriority	Green	Any valid DECwindows named color
VCS$ENS.Window1 VCS01	LightGray	Any valid DECwindows named color
VCS$ENS.Window1 X	Center	"Center" or any valid Screen Pixel Value
VCS$ENS.Window1 Y	Center	"Center" or any valid Screen Pixel Value
VCS$ENS.Window1 Width	800	Any valid Screen Pixel Value
VCS$ENS.Window1 Height	200	Any valid Screen Pixel Value
VCS$ENS.Window1 Font	Fixed	Any Font valid for your Server
VCS$ENS.Window1 Bell	25	Any Value in the range 0 to 100

These DECwindows resources are case sensitive. The following example, taken from the DECW$XDEFAULTS.DAT file, shows how you should enter these values. Note that a space separates the class names.

```
!
! ENS Window1
!
VCS$ENS.Window1 EnsBackground:          NavyBlue
VCS$ENS.Window1 EnsForeground:          SkyBlue
VCS$ENS.Window1 VCS100:                 Blue
VCS$ENS.Window1 VCS99:                  White
VCS$ENS.Window1 HighPriority:           Red
VCS$ENS.Window1 MediumPriority:         Yellow
VCS$ENS.Window1 LowPriority:            Green
VCS$ENS.Window1 VCS01:                  LightGray
VCS$ENS.Window1 X:                      Center
VCS$ENS.Window1 Y:                      Center
VCS$ENS.Window1 Width:                  800
VCS$ENS.Window1 Height:                 200
VCS$ENS.Window1 Bell:   1
```

10.3 Adding Your Own Action Routines

ENS allows you to add your own action routines. These may be in the form of an executable image, a DCL procedure, or a DCL command. This section describes parameter values and how they are passed to the action routine.

There are two formats that ENS can use for passing parameters to your program: a data structure or an ASCII string. Table 10–3 outlines the parameter names and their values.

Table 10–3 ENS Parameter Names and Definitions

Name	Definition
<P1>	System where the event occurred
<P2>	Event name
<P3>	Event priority
<P4>	Event text string
<P5>	Time of the event
<P6>	Time stamp in the console file
<P7>	Event type (SCAN, VCS, or ENS)
<P8>	Information field of the event definition
<PARAMS>	Notification destination
<STRUCTURE>	All the above in a formatted data structure. Used by images only.

10.3.1 Command Procedure Action Routines

Command procedures are the easiest to implement. The following example shows how parameters can be used.

```
ENS EDIT>ADD ACTION MAIL - Return
_ENS EDIT>/COMMAND=@VCS$ENS:SEND_MAIL.COM - Return
_ENS EDIT>/PARAMETERS="<P1> <P2> <P3> ""<P4>"" <PARAMS>" Return
```

The parameters used in this example are the name of the system, the event name, the event priority, and the event text. You can see that this is a command procedure because the command value is preceded by @. The entire parameter list is enclosed in quotation marks and each parameter is separated by a space. Parameter P4, the event text, is enclosed in two sets of quotation marks. The two sets of quotation marks indicate that this parameter may contain embedded blanks and, therefore, will be passed as a quoted string to the command procedure. <PARAMS> is data defined in the event definition which will be passed along to the action routine. In this case, <PARAMS> is the name or names of the account that is to receive mail.

An event definition might look like the following example:

```
ENS EDIT>ADD EVENT REBOOT - Return
_ENS EDIT>/ACT1 = (NAME = MAIL, DEST = (VOBA,STEVENS) )- Return
_ENS EDIT>/SYSTEM = (ALPHA, BETA, GAMMA) Return
```

In this example, the destination "VOBA,STEVENS" is passed along to the action routine called MAIL, substituting <PARAMS> with "VOBA,STEVENS". If there is an embedded space between the two names, then you must enclose the parameter in two sets of quotation marks the same way <P4> is referenced in the previous example. These two names are passed to the command procedure MAIL. Within the command procedure, it is the fifth parameter and is referenced as P5.

A process is then spawned to execute the command procedure. Once the command procedure has finished, the spawned process will cease to exist.

10.3.2 Image Action Routines

Instead of executing a command procedure, you may want to execute a program. This cuts down on the demand for system resources, providing faster notification. When you write your program, design it to stay running after the first event has occurred.

The passing of the parameter values is done through mailboxes. The process name is the image name, several spaces, and a unique number identifier. You can verify that the program is running by entering the VMS command:

```
$ SHOW SYSTEM Return
```

10.3.2.1 Using Mailboxes ENS creates a permanent mailbox when the first event is delivered from VCS. The name of this mailbox is the one you specify when adding the action routine to the ENS data file. If no name is specified, the process name is used. In your program, you must open a channel to the mailbox using the VMS SYS$CREMBX system service and issue QIOs to read the mailbox. There are three types of mailbox messages. The following table defines each type:

Table 10–4 ENS Mailbox Message Types

Type	Definition
VCS_ENS$K_SCAN_EVENT	An event detected by the the VCS Data Scanner.
VCS_ENS$K_VCS_EVENT	An event such as reconfigure or shutdown.
VCS_ENS$K_CONTROL_ EVENT	ENS control message such as startup or shutdown

You must be sure that your program is functioning correctly when using mailboxes. Otherwise, if you do not read the mail, the mailbox will fill up and create a bottleneck for ENS.

The format of the data that ENS writes to the mailbox can be in either ASCII text or in a binary encoded structure format. These two methods are described in the following section.

10.3.2.2 Passing a Parameter String to an Image

The entire list of events can be passed to your program as an ASCII string by way of a mailbox. Delimiters are used to separate the component fields. The delimiter can be any character. Within your program, you must dissect the component fields by searching the string for the delimiter character you specified. The following example shows how to enter an action routine using an encoded ASCII string.

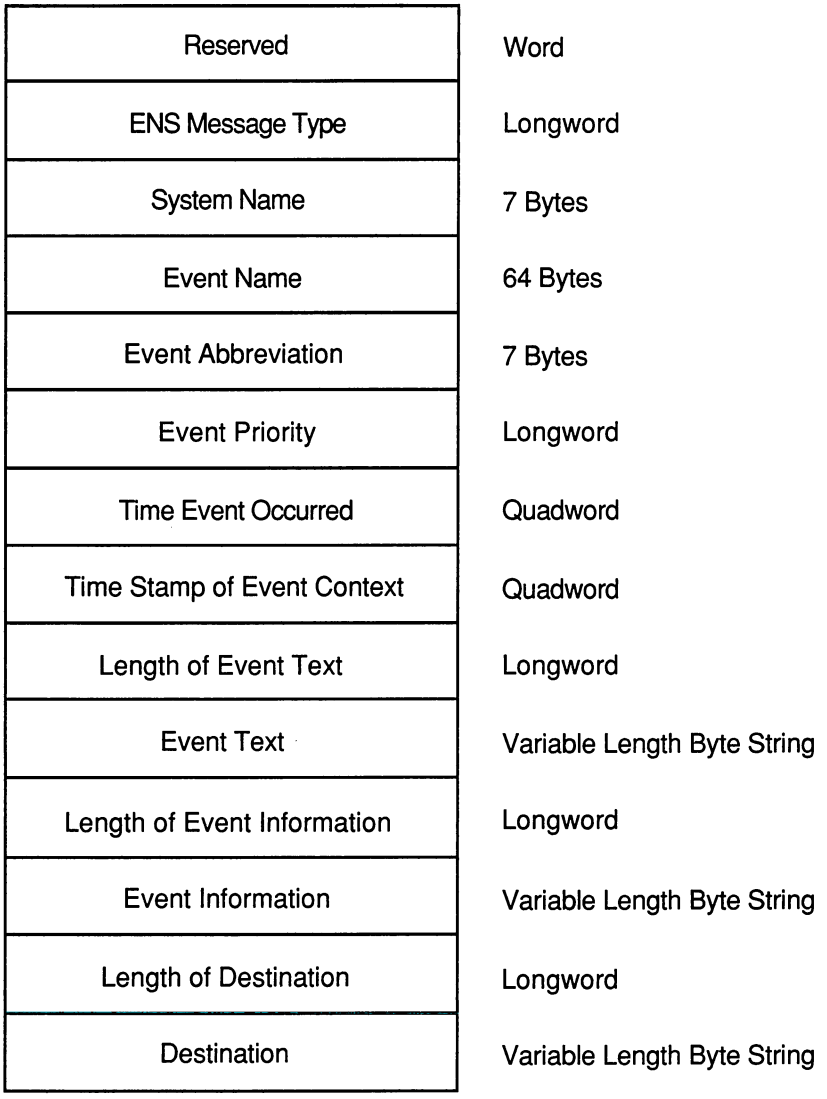
```
ENS EDIT> ADD ACTION BEEPER - Return
_ENS EDIT> /IMAGE=SYS$SYSTEM:BEEPER.EXE - Return
_ENS EDIT> /PARAMETERS=<P1>#<P2>#<P3>#<P4>#<PARAMS># - Return
_ENS EDIT> /MAILBOX = BEEPER$MBX Return
```

In this example, the parameters are separated by the # character as a delimiter.

10.3.2.3 Passing a Parameter Structure to an Image

The second method of passing the parameter values to your program is with a defined data structure. A structure is also passed by way of a mailbox. Figure 10–4 illustrates the ENS mailbox structure.

Figure 10–4 ENS Mailbox Structure



The following example shows how to define this method in the datafile.

```
ENS EDIT> ADD ACTION WINDOW/MAILBOX/UNIQUE - Return
_ENS EDIT> /IMAGE=SYS$SYSTEM:VCS$ENS_WINDOW - Return
_ENS EDIT> /PARAMETERS=<STRUCTURE> Return

ENS EDIT> ADD EVENT SYSMGR - Return
_ENS EDIT> /ACT1 = (NAME=WINDOW, DEST=ALPHA,BETA) Return
```

In this example, the parameters are passed in a structure because the parameter qualifier value, <STRUCTURE>, was specified.

Also note that the unique qualifier in the action means that a new process will be started for each different destination. In this case, the event definition specifies two destinations, ALPHA and BETA. The action routine creates two processes, one for each destination. The process names for ALPHA and BETA would be as follows:

```
WINDOW        01
WINDOW        02
```

10.3.2.4 Testing and Debugging an Action Routine After you have developed a new action routine, you may want to run a version of the program which was compiled and linked with the debugger so that you can step through the code as it is executed. You can do this by putting the program where ENS does not expect to find it. ENS will, however, continue to put event messages in the mailbox. By running your program, you can now open and read the mailbox.

To verify that ENS has created a mailbox, you can enter the following DCL command:

```
$  SHOW LOGICAL mailbox-name Return
```

10.4 ENS Editor Commands

The ENS Editor enables you to add, modify, and delete entries in the ENS data file.

To start the ENS Editor, enter the following command at the DCL prompt:

```
$ RUN SYS$SYSTEM:VCS$ENS_EDITOR Return
```

Following are the ENS Editor commands:

- ADD ACTION
- ADD EVENT
- COPY ACTION
- COPY EVENT
- DELETE
- EXIT
- HELP
- MODIFY ACTION
- MODIFY EVENT
- SHOW

ADD

Adds an event or action entry to the ENS data file.

Format

ADD { **ACTION** *action-name* / **EVENT** *event-name* } *(/qualifier . . .)*

Description

Use the ADD command to add a new action or event entry to the ENS data file. Event entries can be duplicated. Action entries must be unique.

ADD ACTION

Adds a new action to the ENS data file. An action describes the routine to be executed when an event is detected by VCS.

Format

ADD ACTION *action-name (/qualifier . . .)*

Parameters

action-name
is the name of the action. The name must be from 1 to 16 characters in length.

Qualifiers

/DESCRIPTION=text
This is a quoted text string of up to 40 characters describing what the action does. The default is the action name.

/COMMAND=DCL-command
The /COMMAND qualifier defines the DCL command to be executed when this action is triggered. The /COMMAND qualifier is not compatible with the /IMAGE and /MAILBOX qualifiers.

/IMAGE=program-name
The /IMAGE qualifier defines the name of the program to be invoked when this action is triggered.

/MAILBOX=mailbox-name
This qualifier designates a mailbox to pass information to an action routine defined by the /IMAGE qualifier and optionally specifies the mailbox name. If the mailbox name is omitted, the process name, prefixed by **VCS$ENS_**, is used as the mailbox name. The /MAILBOX qualifier is used only with the /IMAGE qualifier.

/UNIQUE
/NOUNIQUE (D)
The /UNIQUE qualifier, when used with the /IMAGE qualifier, causes ENS to start a separate process for each destination parameter defined in the event entry. If the /NOUNIQUE qualifier is specified, only one process is started and all event information is passed to that process, regardless of the destination parameters defined.

The default is /NOUNIQUE.

/PARAMETERS=parameter-list
This qualifier specifies which information is passed to the action routine and how the information should be presented. The following substitution parameters can be used to specify the parameter qualifier for an action entry.

Name	Definition
<P1>	System where the event occurred
<P2>	Event name
<P3>	Event priority
<P4>	Event text string
<P5>	Time of the event
<P6>	Time stamp in the console file
<P7>	Event type (SCAN, VCS, or ENS)
<P8>	Information field of the event definition
<PARAMS>	Notification destination
<STRUCTURE>	All the above in a formatted data structure. Used by images only.

ADD ACTION

Description

Use the ADD ACTION command to specify which image to run or command procedure to execute.

When VCS detects an event, it passes the event on to ENS. ENS then looks in its database for the presence of this event. If the event is present, ENS activates the action defined within the ENS event definition.

The image is not activated until the first event is detected. Multiple actions with the same name are not allowed.

Examples

1

```
ENS EDIT>  ADD ACTION MAIL/DESCRIPTION="Send a mail message" - Return
_ENS EDIT> /COMMAND=@SEND_MAIL.COM - Return
_ENS EDIT> /PARAMETERS="<P1> <P2> <P3> "<P4>"" Return
```

Note that the parameters are enclosed in quotes. When ENS triggers this action a detached process is created to process the following DCL command:

```
$ @SEND_MAIL.COM system event priority "text of event"  Return
```

2

```
ENS EDIT> ADD ACTION BRDCST/DESCRIPTION="Broadcast an event" - Return
_ENS EDIT> /IMAGE=VCS$ENS:VCS$ENS_BRDCST.EXE" - Return
_ENS EDIT> /MAILBOX - Return
_ENS EDIT> /PARAMETERS = <P1>#<P2>#<P3>#<P5>#<P4>#<PARAMS># Return
```

In this example, the parameters are not enclosed in quotes but, are separated by the # sign. In this case, a # sign must also be added after <PARAMS> because the VCS$ENS_BRDCST.EXE image expects the message in that format. When ENS triggers this action, a detached process is created to run the program.

3
```
ENS EDIT>  ADD ACTION WINDOW/DESCRIPTION="Create a window" - Return
_ENS EDIT>  /IMAGE=ENS_WINDOW.EXE  - Return
_ENS EDIT>  /UNIQUE/MAILBOX - Return
_ENS EDIT>  /PARAMETERS = <STRUCTURE> Return
```

In this example, ENS has created a unique process for each destination specified in the event entry. Each process executes the VCS$ENS_WINDOW image. The entire event structure for each process is then passed by a separate mailbox.

ADD EVENT

Adds to the ENS data file a new entry to be triggered when a message is detected on the console of a serviced node.

Format

ADD EVENT *event-name (/qualifier . . .)*

Parameters

event-name
This is the name of the event. The name must be from 1 to 16 characters in length. Use an asterisk (*) to indicate all events.

Qualifiers

/DESCRIPTION=text
This is a quoted text string of up to 40 characters in length describing the event. The default is a null string.

/INTERVAL=delta-time
This is a VMS delta time which specifies the minimum time period allowed between notifications of the same event details. The default is 00:00:00.00

/SYSTEM=node-name
The /SYSTEM qualifier indicates that the event is to an action only on the selected nodes. The default indicating all systems is *.

/ACTn=(NAME=action-name, DEST=user-data)
An action is the response you want the system to take when the occurrence of an event is detected. A maximum limit of eight actions (ACT1-8) per event entry is set, but the number of entries you can specify for the same event has no limit.

NAME=action-name is the name of a previously defined action entry in the ENS database.

DESTINATION=user-data is the data passed as the <PARAMS> field to the action routine. This field can be a list of destinations, however, quotation marks may be required if embedded spaces exist within any of the destinations.

Description

These are the events you want to trap and have an event notification action performed on. The event must also be defined in the VCS configuration file, VCS$CONFIGURATION.DAT. Multiple definitions of the same event are not allowed.

Examples

1
```
ENS EDIT>  ADD EVENT SYSMGR /SYSTEM=(PUGH,TADLEY) - Return
_ENS EDIT>   /ACT2 = (NAME=MAIL, DEST=VOBA) Return
```

This example initiates an action if the event SYSMGR occurs on systems PUGH or TADLEY.

2
```
ENS EDIT>   ADD EVENT SECURE /SYSTEM=(NODEA,NODEB) - Return
_ENS EDIT>   /ACT1=(NAME=WINDOW1,DEST="MUTLEY YOSSAM") Return
```

This example runs the action Window1 on nodes MUTLEY and YOSSAM when the event SECURE occurs.

3
```
ENS EDIT>  ADD EVENT SYSMGR/SYSTEM = (NODEA, NODEB) - Return
_ENS EDIT>   /ACT2=(NAME=WINDOW,DEST=MUTLEY,YOSSAM) Return
```

This example causes the action Window to display windows on nodes MUTLEY and YOSSAM when the event SYSMGR occurs.

COPY

Copies an event or action entry to a new name in the ENS data file.

Format

COPY { **ACTION** *source-name destination-name* } *(/qualifier . . .)*
{ **EVENT** *source-name destination-name* }

COPY ACTION

Copies an action entry in the ENS data file.

Format

COPY ACTION *source-action-name dest-action-name (/qualifier . . .)*

Parameters

source-action-name
is the name of the action to be copied in the ENS data file. The name must be from 1 to 16 characters in length.

dest-action-name
is the name of the action to be copied to in the ENS data file. The name must be from 1 to 16 characters in length.

Qualifiers

/DESCRIPTION=text
This is a quoted text string of up to 40 characters describing what the action does. This field may be used for encoding instructions that tell the recipient of the notification how to respond to a particular VCS action.

/COMMAND=action
The /COMMAND field defines which DCL command to execute when triggered by an action. The /COMMAND is not compatible with the /IMAGE and /MAILBOX qualifiers.

/IMAGE=program-name
The /IMAGE field defines the name of the image to be invoked when the action is triggered.

COPY ACTION

/MAILBOX=mailbox-name
This qualifier designates a mailbox to pass information to an action routine defined by the /IMAGE qualifier and optionally specifies the mailbox name. If the mailbox name is omitted, the process name, prefixed by **VCS$ENS_**, is used as the mailbox name. The /MAILBOX qualifier is used only with the /IMAGE qualifier.

/UNIQUE
/NOUNIQUE (D)
The /UNIQUE qualifier, when used with the /IMAGE qualifier, causes ENS to start a separate process for each destination parameter defined in the event entry. If the /NOUNIQUE qualifier is specified, only one process is started and all event information is passed to that process, regardless of the destination parameters defined.

The default is /NOUNIQUE.

/PARAMETERS=parameter-list
This qualifier specifies which information is passed to the action routine, and how the information should be presented.

Description

The following substitution parameters can be used to specify the parameter qualifier for an action entry.

Name	Definition
<P1>	System where the event occurred
<P2>	Event name
<P3>	Event priority
<P4>	Event text string
<P5>	Time of the event
<P6>	Time stamp in the console file
<P7>	Event type (SCAN, VCS, or ENS)
<P8>	Information field of the event definition
<PARAMS>	Notification destination
<STRUCTURE>	All the above in a formatted data structure. Used by images only.

Example

```
ENS EDIT> COPY ACTION MAIL MAIL1/DESCRIPTION="Send a mail message" - Return
_ENS EDIT> /COMMAND=@SEND_MAIL.COM  Return
```

This example causes the action MAIL to be copied to MAIL1 and modifies the description and command fields.

COPY EVENT

Copies an event entry in the ENS data file.

Format

COPY EVENT *source-event-name dest-event-name (/qualifier . . .)*

Parameters

source-event-name
This is the name of the event to be copied. The name must be from 1 to 16 characters in length.

dest-event-name
This is the name of the new event. The name must be from 1 to 16 characters in length.

Qualifiers

/DESCRIPTION=text
This is a quoted text string of up to 40 characters describing the event.

/INTERVAL=delta-time
This is a VMS delta time which specifies the minimum time period allowed between notifications of the same event details.

/SYSTEM=node-name
The /SYSTEM qualifier indicates that the event is to run an action only on the selected nodes.

/ACTn=(NAME=action-name, DEST=user-data)
An action is the response you want the system to take when the occurrence of an event is detected. A maximum limit of eight actions (ACT1-8) per event entry is set, no limit to the number of entries you can specify for the same event is set. No checking is done to verify that the action is on file.

NAME=action-name is the name of a previously defined action entry in the ENS database.

DESTINATION=user-data is the data passed as the <PARAMS> field to the action routine. This field may be a list of destinations, however, quotation marks may be required if embedded spaces exist within any one of the destinations.

Description

These are the events you want to trap and have an action performed on.

Example

```
ENS EDIT> COPY EVENT SYSMGR SYSMGR1/SYSTEM=(HAMPS) Return
```

In this example, event definition SYSMGR is copied to SYSMGR1 and modifies the system parameter to HAMPS.

DELETE

Deletes an existing entry in the ENS data file.

Format

DELETE { **ACTION** *action-name* | **EVENT** *event-name* }

Parameters

action-name
is the name of the action being deleted. You may use the wildcard character, asterisk (*), to indicate all existing action entries being deleted from the ENS data file.

event-name
is the name of the event being deleted. You may use the wildcard character, asterisk (*), to indicate all existing event entries being deleted from the ENS data file.

Description

Use the DELETE command to delete an existing entry from the ENS database.

Examples

1
```
ENS EDIT> DELETE ACTION ALPHA Return
_ENS EDIT>
```

This command causes the action entry ALPHA to be deleted from the ENS data file.

2
```
ENS EDIT> DELETE EVENT * Return
_ENS EDIT>
```

This example causes all event entries to be deleted from the ENS data file.

EXIT

Terminates an editing session.

Format

EXIT

Description

The EXIT command terminates an editing session and returns you to the DCL command level.

HELP

Invokes the ENS Editor online help facility.

Format

HELP *(topic)*

Parameters

topic
is the name of the command for which you requested help. If you omit the topic, the ENS Editor displays a list of available topics.

Description

The help facility displays information about requested ENS Editor topics. At the **ENS EDIT>** prompt, you can display a list of topics by typing HELP and pressing Return. The ENS Editor responds by displaying a brief description of how to use the help facility, followed by a list of topics for which help is available. The prompt is changed to **Topic?**. In response to the **Topic?** prompt, you can:

- Enter one of the topic names from the list.
- Press Return one or more times to exit, or press Ctrl/Z to immediately exit from from the help facility.
- Enter a question mark (?) to see a list of topics.
- Enter an asterisk (*) to display all available information from the help facility.
- Enter a topic immediately followed by an ellipsis (. . .) to display all of the information on a topic. For example, ADD . . . displays all of the information available for the ADD command.

You can abbreviate any topic name, although ambiguous abbreviations result in all matches being displayed.

Example

```
ENS EDIT> HELP ADD Return
```

When you enter this command, the ENS Editor displays a description of the ADD command.

MODIFY

Modifies an existing entry in the ENS data file.

Format

MODIFY { **ACTION** *action-name* | **EVENT** *event-name* }

Description

Use the MODIFY command to edit an existing entry in the ENS data file.

MODIFY ACTION

This command modifies an action entry in the ENS data file.

Format

MODIFY ACTION *action-name (/qualifier . . .)*

Parameters

action-name
is the name of the action. The name must be from 1 to 16 characters in length.

Qualifiers

/DESCRIPTION=text
This is a quoted text string of up to 40 characters in length describing what the action does. The default is a null string.

/COMMAND=DCL-Command
The /COMMAND qualifier defines the DCL command to be executed when this action is triggered. The /COMMAND qualifier is not compatible with the /IMAGE and /MAILBOX qualifiers.

/IMAGE=program-name
The /IMAGE qualifier defines the name of the image to be invoked when this action is triggered.

/MAILBOX=mailbox-name
This qualifier designates a mailbox to pass information to an action routine defined by the /IMAGE qualifier and optionally specifies the mailbox name. If the mailbox name is omitted, the process name, prefixed by **VCS$ENS_**, is used as the mailbox name. The /MAILBOX qualifier is used only with the /IMAGE qualifier.

MODIFY ACTION

/UNIQUE
/NOUNIQUE (D)
The /UNIQUE qualifier, when used with the /IMAGE qualifier, causes ENS to start a separate process for each destination parameter defined in the event entry. If the /NOUNIQUE qualifier is specified, only one process is started and all event information is passed to that process, regardless of the destination parameters defined.

The default is /NOUNIQUE.

/PARAMETERS=parameter-list
This qualifier specifies which information is passed to the action routine, and how the information should be presented.

Description

The following substitution parameters can be used to specify the parameter qualifier for an action entry.

Name	Definition
<P1>	System where the event occurred
<P2>	Event name
<P3>	Event priority
<P4>	Event text string
<P5>	Time of the event
<P6>	Time stamp in the console file
<P7>	Event type (SCAN, VCS, or ENS)
<P8>	Information field of the event definition
<PARAMS>	Notification destination
<STRUCTURE>	All the above in a formatted data structure. Used by images only.

Example

```
ENS EDIT> MOD ACTION BRDCST /DESC="Broadcast an EVENT to a Terminal" - Return
_ENS EDIT> /IMAGE=SYS$SYSTEM::VCS$ENS_BRDCST.EXE - Return
_ENS EDIT> /MAILBOX=ENS_BRDCST_MBX Return
```

This is an example of how to modify the /IMAGE and /MAILBOX qualifiers using an action called BRDCST.

MODIFY EVENT

Modifies an event entry in the ENS data file.

Format

MODIFY EVENT *event-name (/qualifier . . .)*

Parameters

event-name
is the name of the event. The name must be from 1 to 16 characters in length.

Qualifiers

/DESCRIPTION=text
This is a quoted text string of up to 40 characters describing the event. The default is a null string.

/INTERVAL=delta-time
This is a VMS delta time which specifies the minimum time period allowed between notifications of the same event details.

/SYSTEM=node-name
The /SYSTEM qualifier indicates that the event is to run an action only on the selected nodes.

MODIFY EVENT

/ACTn= { ***(NAME=action-name, DEST=user-data)*** / ***REMOVE*** }

An action is the response you want the system to take when the occurrence of an event is detected. A maximum limit of eight actions (ACT1-8) per event entry is set. No limit to the number of entries you can specify for the same event is set.

NAME=action-name is the name of a previously defined action entry in the ENS database.

DESTINATION=user-data is the data passed as the <PARAMS> field to the action routine. This field may be a list of destinations, however, quotation marks may be required if embedded spaces exist within any one of the destinations.

To remove an action, specify **ACTn=REMOVE**.

Description

The MODIFY EVENT command allows you to edit the information of an existing entry describing an event.

SHOW

Displays information about a specific entry in the ENS data file.

Format

SHOW { **ACTION** *action-name* } { **EVENT** *event-name* }

Parameters

action-name
is the name of the action. You can use the wildcard character, asterisk (*), to indicate all existing actions in the ENS data file.

event-name
is the name of the action. You can use the wildcard character, asterisk (*), to indicate all existing actions in the ENS data file.

Qualifiers

/FULL
This qualifier shows the full description for the entry.

/BRIEF
This is the default qualifier and shows a brief description for the entry.

/OUTPUT=filespec
This qualifier redirects the output of the SHOW command the the specified file.

Description

Use the SHOW command to display information about any current definition of the various components in the ENS data file. The /FULL and /BRIEF qualifiers are meaningful only when they are used to display the information about the action and event entries created in the data file by the ADD command.

Examples

1 ENS EDIT> SHOW ACTION */OUTPUT=ENS.LIS Return

This example displays information about all actions. The information is directed to the ENS.LIS file.

2 ENS EDIT> SHOW ACTION */FULL Return

This example shows actions with the /FULL qualifier.

```
BRDCST              Broadcast message to User or Terminal
    Command : ENS:ENS_BRDCST
 Parameters : <P1>#<P2>#<P3>#<P5>#<P4>#<PARAMS>#
    Mailbox : ENS_BRDCST_MBX
WINDOW              DECWindows Message Box
    Command : ENS:ENS_WINDOW
 Parameters : <P1>#<P3>#<P4>#<P5>#<PARAMS>#
    Mailbox : ENS_WINDOW_MBX
WINDOW1             Old Window System
    Command : @ENS:ENS_WINDOW1.COM
 Parameters : <P1> <P2> <P3> "<P4>" "0 00:10:00.00" "<PARAMS>"
    Mailbox : ** NONE **
```

A

Log Files

This appendix describes the VCS console and event log files and how to manage them.

A.1 Introduction

The VCS log files contain the complete date encoded into their names. VCS uses the format of VCS$*yyyy_mm_dd*.LOG and VCS$EVENT_*yyyy_mm_dd*.LOG to encode the console and event log files respectively. For example, **VCS$1988_09_07.LOG** and **VCS$EVENT_1988_09_07.LOG** are the console and event log files for September 7, 1988. VCS defines the system-wide logical name VCS$LOG which points to the directory where the log files are created. The Monitor Interface SHOW DIRECTORY command displays the names of the log files as well as their location.

VCS creates a new console log file each day, beginning at midnight. A single console log file contains all console data from all serviced nodes for that day. A single event log file contains all VCS events and console events that were detected on all serviced nodes' consoles for that day.

Digital suggests that you, the system manager, develop a schedule for regularly backing up log files generated by VCS. The number of serviced nodes and the amount of console line data normally generated by these nodes will affect the backup schedule. To back up the log files, use standard VMS backup procedures. You can back up the files to tape. Alternatively, if you have the DECnet option installed on your VCS host system, you can back up the files to a disk volume on another node in the network. After backing up your oldest log files from the disk, you can delete them. Try to achieve a balance between maintaining adequate disk space and keeping a reasonable number of files on disk. The

more log files you can retain on the disk, the more data users can access for retrieval of historical console data.

Backing up the VCS log files and deleting the oldest log files from the disk are optional tasks. VCS will automatically delete the oldest log files when free space on the disk, where the directory VCS$LOG is located, gets below a specified threshold.

When the disk has approximately 2000 blocks or 2% of the total disk blocks left (whichever is greater), VCS displays the following message in the I/O Data Logger process log file SYS$MANAGER:VCS$IODL_OUTPUT.LOG:

```
%VCS-W-FULLWARN, Disk full warning, please free space on logging disk
```

When you receive this warning, immediately back up the oldest log files before VCS deletes them. After backing up the oldest log files, you can delete them. The disk full warning threshold can be adjusted using the Configuration Editor (see Chapter 3 and Chapter 4). You may also automate the backup process by specifying a log file archival procedure using the Configuration Editor.

If you do not free additional disk space, VCS begins to delete the console and event log files, starting with the oldest ones, until sufficient free disk space is recovered. VCS also displays the following message in the I/O Data Logger process log file:

```
%VCS-I-FULLDELETE, Disk full action taken, log file filename deleted
```

Filename is the actual name of the log file that was deleted. This threshold can be adjusted using the Configuration Editor.

Note *VCS places all characters received on a node's console line into the log file. If a program produces many repetitious messages on the console line, VCS continues to write these messages into the log file as long as there is adequate disk space. To prevent the disk from becoming full and to help prevent VCS from deleting log files, you can use the Monitor Interface DISABLE command to temporarily disconnect a serviced node that is producing repetitious messages. After you fix the problem, you can reconnect the serviced node to VCS with the Monitor Interface ENABLE command.*

A.2 Log File Analysis

The Review Interface, invoked by either the VCS REVIEW or the Console Monitor Interface REVIEW command, provides a convenient way to retrieve the console and event data from the log files. The data can be selectively retrieved and displayed or written to a flat ASCII file.

VCS provides the file VCS$LOG_RECORDS_DESCRIPTION.COM in the directory VCS$EXAMPLE, that has the Datatrieve definitions of the log records. You can reference the information in this file for developing your applications to analyze the log files. The remaining material in this appendix is a more formal description of that information.

A.3 Console and Event Log File Record Formats

Each VCS log file is a single-key RMS indexed sequential file. The log file consists of a series of records, each of which contains a type field in the record header that indicates both the type and format of the information contained in that record. The primary time field in the record header, which indicates when the record is written to the log file, is the RMS index key. Figure A–1 shows the information contained in the header of each record.

Figure A–1 Record Header

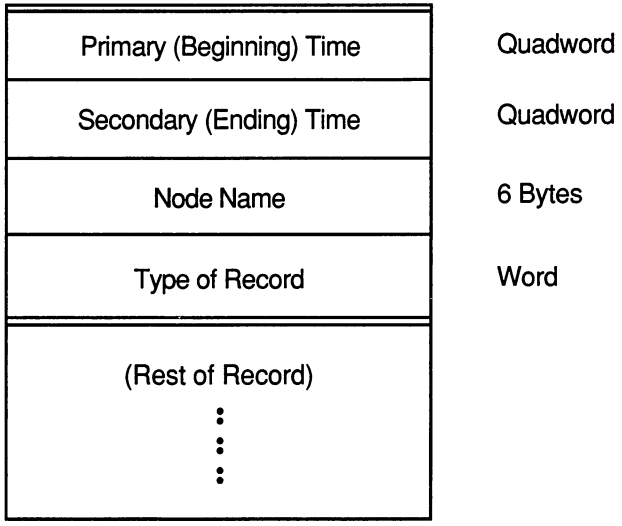


Table A–1 lists and describes each field of the Record Header:

Table A–1 Record Header Fields

Record Field	Type	Field Description
Primary (Beginning) Time	Quadword	The time of reference for the record. In most cases, this indicates the time the record is created. This is the RMS Index Key Field.
Secondary (Ending) Time	Quadword	The time of completion of the record or the start time for some other record or null.
Node Name	6 Bytes	The node name associated with the record. For log data and user output data, this field contains the name of the target node. For the other records, it contains the acronym VCS.
Type of record	Word	The type of information contained and the format of the record. See Table A–2.

Table A–2 lists the record types of log file records:

Table A–2 Console and Event Log Record Types

Record Type	Code	Description
VCS$K_VERSION_RECORD	1	Version record that indicates the version of the records in the file. This is the first record of every log file.
VCS$K_ERROR_RECORD†	2	VCS error record that indicates an error occurred in VCS itself.
VCS$K_SERVICE_RECORD†	3	Service initiate/terminate record that indicates the date and time a node is enabled or disabled.
VCS$K_USER_STATUS_RECORD	4	User status change that indicates the date and time a node is enabled or disabled.

†These records are displayed by the Monitor Interface and can be retrieved by the Review Interface. You can only display VCS$K_ERROR_RECORD if the record header node name is a name that is not VCS.

(continued on next page)

Table A–2 (Cont.) Console and Event Log Record Types

Record Type	Code	Description
VCS$K_USER_DATA_RECORD	5	User output to console that contains the text length sent by a VCS user to a node. The primary purpose of this record type is to provide an audit trail of the VCS user's activities.
VCS$K_PARTIAL_DATA_RECORD†	6	Partial Log Data Record that contains the **Node to User** text received from a node. The record was created because no line terminator was received within *n* seconds, where *n* is a VCS-defined threshold. This threshold is currently defined to be 30 seconds.
VCS$K_COMPLETE_DATA_RECORD†	7	Completion Log Data Record that contains the **Node to User** text received from a node. This record implies the existence of a partial log-data record in a log file other (older) than the current one. The time stamp for this record is always immediately after midnight, 00:00:00.00.
VCS$K_FULL_DATA_RECORD†	8	Full Log Data Record that contains the **Node to User** text received from a node.
VCS$K_VCS_EVENT_RECORD	9	VCS event record that indicates a VCS event.
VCS$K_SCAN_EVENT_RECORD	10	Scan event record that indicates a scan event has been detected by the Data Scanner.

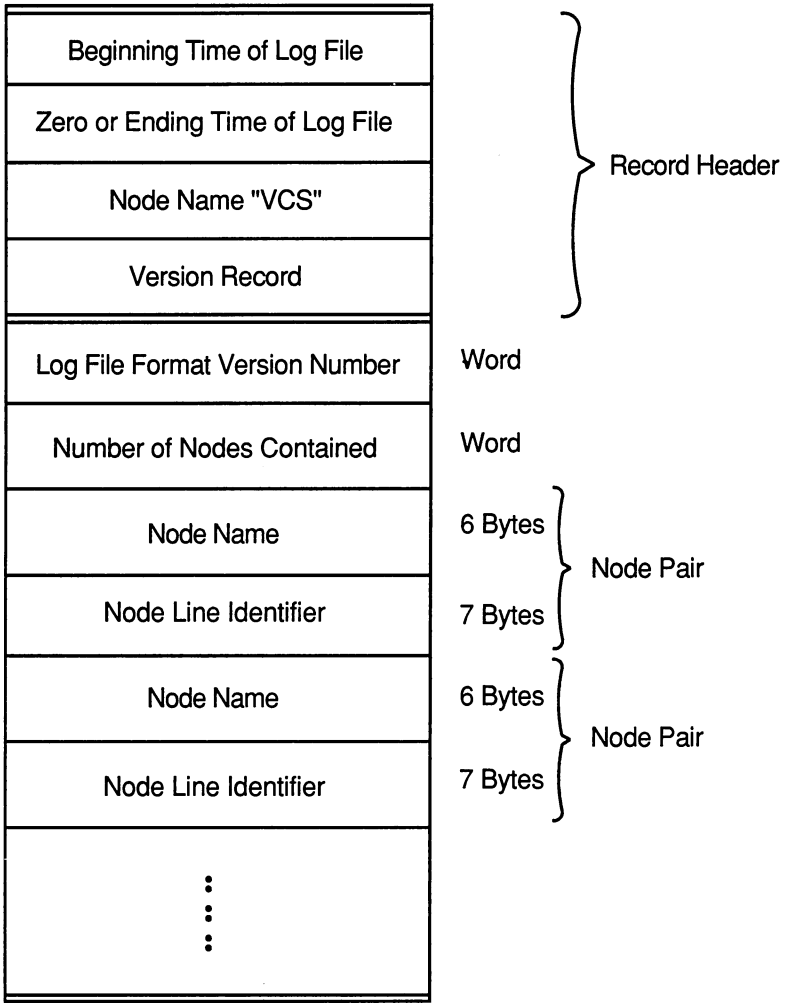
†These records are displayed by the Monitor Interface and can be retrieved by the Review Interface. You can only display VCS$K_ERROR_RECORD if the record header node name is a name that is not VCS.

The following sections describe the log file record formats. The general descriptions include the data types and sizes of the record fields.

A.3.1 Version Record

The version record specifies the version of the format of the records in the file.

Figure A–2 Version Record



VCS writes the version record into the event and console log files whenever the files are opened for writing. Therefore, the version record is written whenever the loggers are started up, and when the logger opens a new log file at midnight. The first record of every log file is always a version record.

The version record includes the current configuration assignments. The information is in a list containing the nodes in the current VCS configuration and the device to which each node is assigned. Because the version record contains the configuration data, a version record is also written when a RECONFIGURE command is issued.

Table A–3 lists and describes each field of the Version Record:

Table A–3 Version Record Information

Record Field	Type	Field Description
Time	Quadword	The beginning time of log file that indicates the startup of VCS for the day. If VCS is running continuously, the time is expected to be at midnight; if VCS is started later, this time indicates when the I/O Data Logger process began service. The RECONFIGURE command also writes version records therefore, time also represents the reconfigure time.
Secondary (ending) time	Quadword	The time this log file was closed. If zero, this indicates that VCS provided service for only part of the day.
Log-File Format Version Number	Word	Version number is used for compatibility because it indicates the format of all records in the log for the day.
Number of nodes contained	Word	The number of nodes in the configuration. It also indicates the number of node pairs contained in this record.
Node pair	13 Bytes	Node pair contains two pieces of information (one is 6 bytes and the other is 7 bytes), the node name and the device designator for the terminal line that supports the specified node.

A.3.2 VCS Error Record

The VCS error record type indicates an error occurred in VCS.

Figure A–3 VCS Error Record

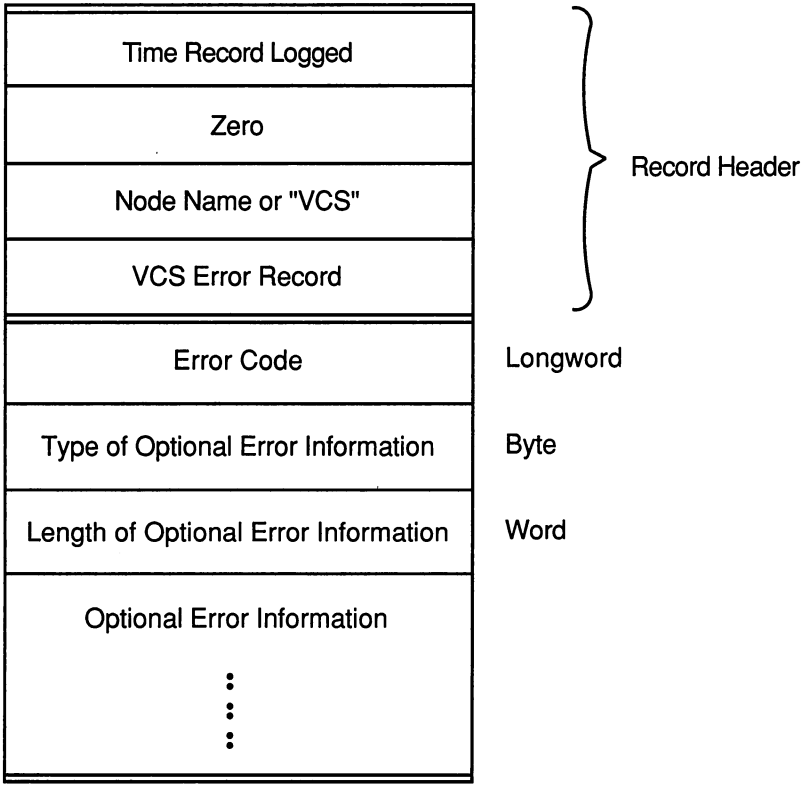


Table A–4 lists and describes each field of the VCS Error Record:

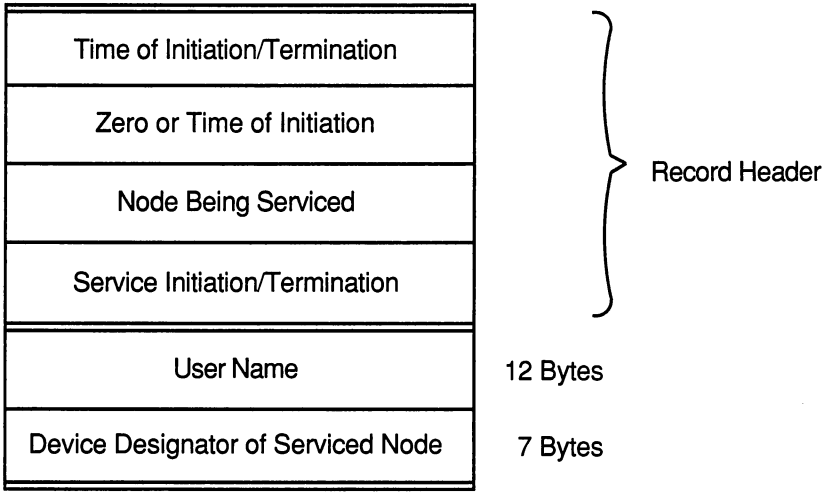
Table A–4 VCS Error Record Information

Record Field	Type	Field Description
Node name or "VCS"	6 Bytes	If node name is the name of a node then the optional error information is an error message. If node name is "VCS", the optional error information can be a message or encoded error information.
Error code	Longword	VMS style error code.
Type of optional error information	Byte	Description of optional error information (binary, characters) contents. This is the VAX Data Type for arguments described in the *Introduction to VAX/VMS System Routines*.
Length of optional error information	Word	The number of bytes in the optional error information field.
Optional error information	Variable Length	Context information further describing the VMS error encountered. Typically this is an error message but, this field can contain encoded error information. (Currently, no encoding scheme has been implemented for VCS specific error information.)

A.3.3 Service Initiation/Termination Record

This record indicates the beginning/ending of VCS service for a particular node. The ENABLE command initiates service and DISABLE terminates service.

Figure A–4 Service Initiation/Termination Record



MR–1778–RA

Table A–5 lists and describes each field of the Service Initiation/Termination Record:

Table A–5 Service Initiation/Termination Record Information

Record Field	Type	Field Description
Time	Quadword	Time of the record.
Secondary time	Quadword	Zero indicates service initiation. A non-zero value indicates service termination, and the secondary time is the time of service initiation.

(continued on next page)

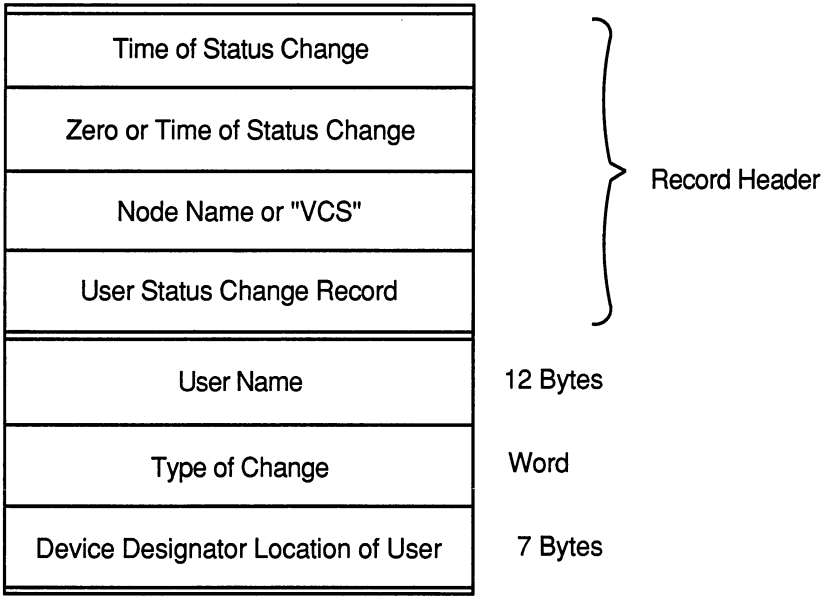
Table A–5 (Cont.) Service Initiation/Termination Record Information

Record Field	Type	Field Description
User name	12 Bytes	The name of the user requesting service initiation or termination.
Device designator used for service	7 Bytes	The terminal line of the node that had its service changed.

A.3.4 User Status Change Record

This record type indicates a user has signed on or off a node, or locked or unlocked a node. A user has signed on/off each time he runs/exits VCS$IFEX (one of the user interfaces).

Figure A–5 User Status Change Record



MR–1779–RA

Table A–6 lists and describes each field of User Status Change Record information.

Table A–6 User Status Change Record Information

Record Field	Type	Field Description
Time	Quadword	This is the time of the record. That is, the time of the sign-on, sign-off, lock or unlock.
Secondary time	Quadword	If this record indicates a sign-on or a lock, the secondary time is zero. If this record indicates a sign-off or an unlock, the secondary time contains the time of the user signon or lock.
Node Name	6 Bytes	If sign-on or sign-off, this contains "VCS"; if this record indicates a lock or unlock, this contains the node name of the node being locked or unlocked.
User Name	12 Bytes	The name of the user changing status.
Type of Change	Word	The specific type of change listed in Table A–7.
Device designator location of user	7 Bytes	The terminal line designator indicating the line used.

Table A–7 User Change Codes

Change Code	Value	Description
VCS$K_UC_SIGNON	1	User has signed on.
VCS$K_UC_SIGNOFF	2	User has signed off.
VCS$K_UC_LOCK	3	User has locked a node.
VCS$K_UC_UNLOCK	4	User has unlocked a node.

A.3.5 User to Node Output Record

This record type contains the text length sent by a user to a node.

Figure A–6 User to Node Output Record

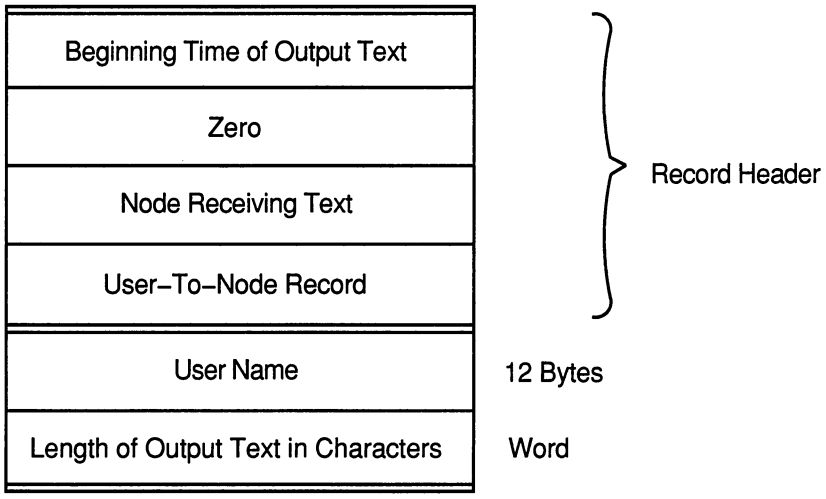


Table A–8 lists and describes each field of the User to Node Output Record:

Table A–8 User to Node Output Record Information

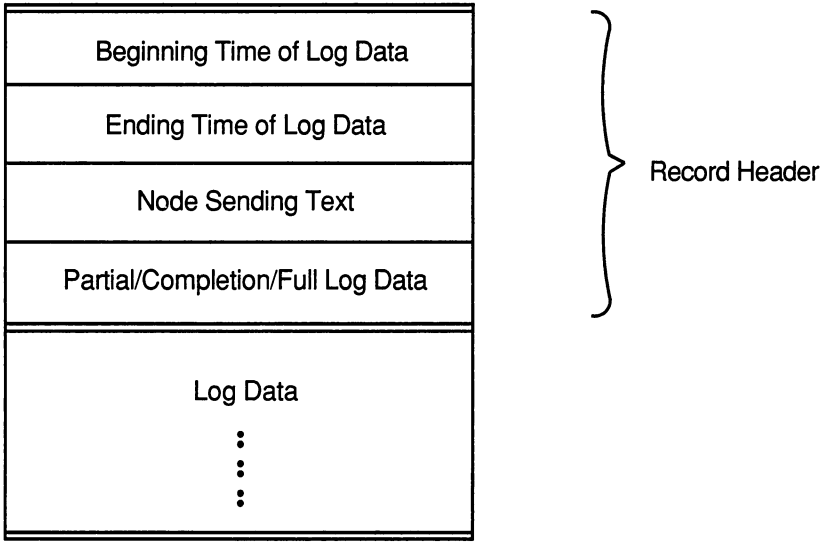
Record Field	Type	Field Description
User name	12 Bytes	The user sending output or commands to the specific node.
Length of text in characters	Word	The number of characters sent to the node.

Note *The output text string is not included in the record for security reasons.*

A.3.6 Log Data Record — (Partial, Complete, or Full)

This record type contains the text received from a node. There is one record for each line of text.

Figure A–7 Log Data Record



MR–1781–RA

Table A–9 lists and describes each field of the Log Data Record:

Table A–9 Log Data Record Information

Record Field	Type	Field Description
Beginning time of log data	Quadword	The I/O Data Logger process tries to save a line of information at a time, therefore, this is the time the first character was detected. The first character is typically the DCL prompt, "$". This time stamp may be displayed by the VIEW, REVIEW and the VCS RECORD commands. Note that the time stamp represents neither the beginning nor the end of the command, but the time that the first character of the line was received by VCS from the console line.
Ending time of log data	Quadword	The time the last character in the line was detected.
Log data	Variable Length	The text string from a node.

In Table A–10, *Real Time* is the time that the first character of a record, *Data Record*, is typed on a console line. The I/O Data Logger process assigns to each Data Record a *Time Stamp Record* and writes both in the log file. The first Data Record, "$", is called a partial line because no carriage-return (C_R) and linefeed (L_F) pair was output within *n* seconds of the first character, "$". The duration *n* (currently defined as thirty seconds) represents the threshold time that the I/O Data Logger process waits before declaring a line to be a partial line and writing it to the log file.

Table A–10 Partial Lines

Real Time	Log File Time Stamp	Log File Data Record
07-MAR-1986/13:00	07-MAR-1986/13:00:00	$
08-MAR-1986/14:00	07-MAR-1986/00:00:00.00	$ directory C_R L_F
	08-MAR-1986/14:00:01.20	Directory DISK:[USER]
	08-MAR-1986/14:00:03.00	FILE.TYPE;1 . . .

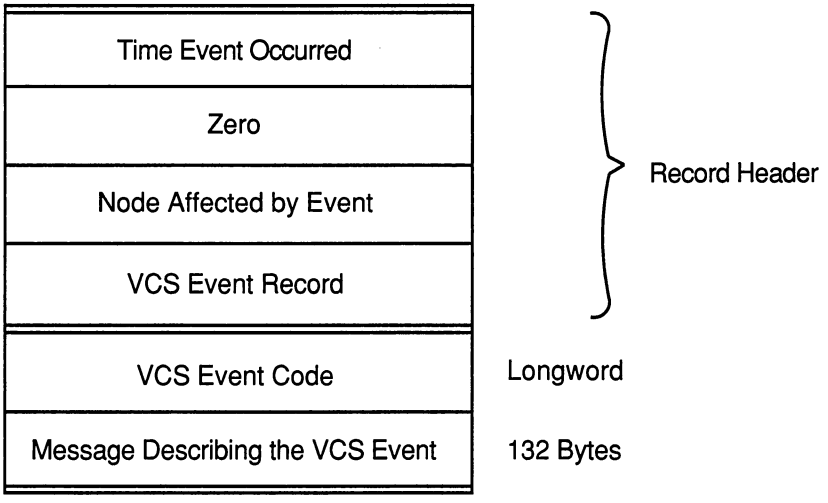
If the line is completed on the same day as the partial line, it is rewritten to the same Data Record record as a Full Data record. If the line is completed after 23:59:59.99 then another Data Record which contains both the partial line and its completion is made in the next day's log file . The time stamp for this type of record is always 00:00:00.00 on the day of the current log file. This is true even when the partial line is completed hours into the day. If there are complete days between the first and last characters of a line then no record is made in the intervening day's log files.

The partial line flag, [PF], and completion line flag, [CF], are appended only to their respective lines when the REVIEW command is invoked with two or more nodes specified.

A.3.7 VCS Event Record

This record type indicates that a VCS event occurred, for example, a VCS user sign-on event.

Figure A–8 VCS Event Record



The following table lists and describes each field of the Event Record:

Table A–11 VCS Event Record Information

Record Field	Type	Field Description
Time event occurred	Quadword	The VCS time stamp associated with the event.
Secondary time	Quadword	Always zero.
Node name	6 Bytes	The node name, if any, affected by the event. For instance, if the event indicates that a system has been locked in the Monitor Interface, the node name is the name of that system. If the event indicates a user has signed on to VCS, then the node name is "VCS".
Type of Record	Word	Event Record.
VCS event code	Longword	The code indicating which event has occurred. Table A–12 lists the VCS event codes.
Message describing the VCS Event	132 Bytes	The message text.

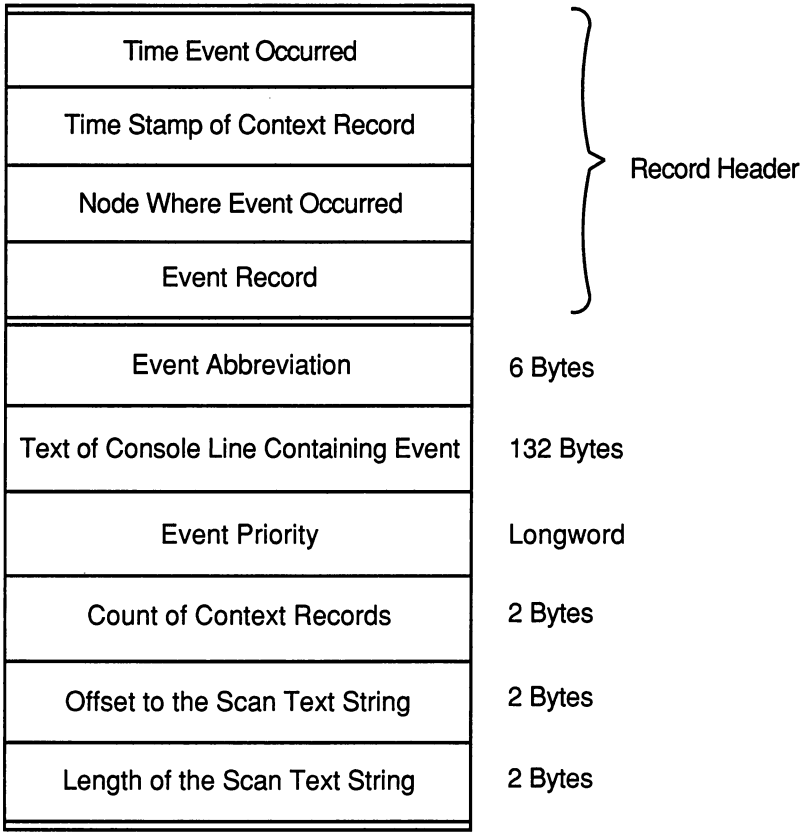
Table A–12 VCS Event Codes

Event Code	Description
VCS$K_DISABLE	Service to a system has been disabled.
VCS$K_ENABLE	Service to a system has been enabled.
VCS$K_ERROR	A VCS error has been detected.
VCS$K_EVENT	An unspecified VCS event has been detected.
VCS$K_LOCK	A system has been locked.
VCS$K_RECONFIG	A VCS reconfiguration has occurred.
VCS$K_TERMINATION	VCS has been shutdown.
VCS$K_UNLOCK	A system has been unlocked.
VCS$K_USRSIGNON	A new user of VCS has signed on.
VCS$K_USRSIGNOFF	A user of VCS has exited.

A.3.8 Scan Event Record

This record type indicates that a scan event occurred.

Figure A–9 Scan Event Record



The following table lists and describes each field of the Scan Event Record:

Table A–13 Scan Event Record Information

Record Field	Type	Field Description
Time event occurred	Quadword	The VCS time stamp associated with the event. Each event has the same time stamp as the one found in the VCS log file for that event.
Time stamp of context record	Quadword	Indicates the time stamp of a record in the VCS log file which is the first record containing the context of this event.
Node Name	6 Bytes	The name of the node on which the event occurred.
Type of Record	Word	Scan Event Record
Event Abbreviation	6 Bytes	Abbreviation of the scan event name.
Text of console line containing event	132 Bytes	The text of the console line which contains this event.
Event priority	4 Bytes	The priority of this event.
Count of context records	Word	Number of context records, including the first one which is specified by the context time stamp above. Using this count, and the time stamp of the context record, all of the context records can be retrieved from the VCS log file.
Offset to the Scan Text String	Word	The offset to the scan text string within the text string of the console line containing the event.
Length of the Scan Text String	Word	The length (in characters) of the scan text string.

B

System Messages and Error Recovery Procedures

B.1 General Instructions

This appendix lists the messages you may see displayed on the terminal screen while working with a VCS interactive interface, or written to a VCS process log file. These messages have the following format:

```
%VCS-Level-Ident, Text
```

A percent sign (%) prefixes the first message issued, and a hyphen (-) prefixes each subsequent message. The *level* field, represented by a character, indicates the severity level and can be one of the following:

Table B–1 Severity Levels

Code	Meaning
S	Success
I	Information
W	Warning
E	Error
F	Fatal

The severity level of a message is included in the status message from the component. You can identify the type of message you have received by its severity level. Success and information messages inform you that the system has performed your request. In some cases, the command processing continues after issuing a success message. In other cases, the message indicates that the command processing is complete.

Warning messages indicate that the command may have performed some, but not all, of your request, and that you may have to verify the command or program output.

Error messages indicate that the output or program result is incorrect, but the system may attempt to continue execution.

Fatal messages indicate that the system cannot continue execution of the request.

The *ident* field is an abbreviation of the message text. All the messages in this appendix are alphabetized according to the *ident* field.

The *text* field is the explanation of the message.

All VCS messages are listed in the next section. If you receive a message that is not listed in this appendix, check the *VAX/VMS System Messages and Recovery Procedures Reference Manual*.

B.2 System Messages and Error Recovery Procedures

ABOPRC, Abort further processing

Severity: Success

Explanation: A function, requiring several steps, has been aborted.

User Action: Take corrective action based on the accompanying message.

ACCVIO, VCS authorized access and privileges violation

Severity: Fatal

Explanation: You are not allowed to access a system serviced by VCS.

User Action: See the system manager to obtain access rights.

ADDED, *identifier* added to configuration file

Severity: Success

Explanation: Identifies that the record has been successfuly added to the configuration file.

User Action: None.

ALLNODES, Only one monitor session allowed for 'all nodes'

Severity: Warning

Explanation: There can be only one Monitor Interface for all systems that is created by clicking on the Monitor command in the VCS icon pop-up Command menu.

User Action: To create another Monitor Interface for all systems, choose the Select All followed by the Monitor from the Command menu.

AMBGSYSTEM, Ambiguous *node-name* specified

Severity: Informational

Explanation: You did not type enough characters to identify the node name.

User Action: Type enough characters to uniquely identify the node.

APPENDFILE, Appending *file-name*

Severity: Informational

Explanation: The specified file is being appended.

User Action: None.

BADCFIL, Failure getting the configuration file name

Severity: Error

Explanation: The request for the name of the currently running configuration file failed.

User Action: Shut down and restart VCS.

BADCONFIG, Configuration file did not load correctly

Severity: Fatal

Explanation: The data in your configuration file is not consistent.

User Action: Your configuration file my have become corrupted. Restore or rebuild the configuration file.

BADDEV, Can not open device *device-name*

Severity: Error

Explanation: The device name to be used as SYS$INPUT and SYS$OUTPUT for a C³ subprocess window cannot be opened. The DECterm window may have been closed before the subprocess started.

User Action: Retry the command.

BADENDDAT, End date:time cannot be before begin date:time

Severity: Error

Explanation: The end date you specified to the REVIEW command precedes the begin date.

User Action: Enter an end date that comes after the begin date.

BADEVELOG, Bad event log file *file-name*

Severity: Error

Explanation: VCS$ELOG notifies that it has encountered a corrupted VCS event log file.

User Action: Shut down VCS. Delete or move the current console and event log files to another directory different from VCS$LOG. Restart VCS.

BADLOG, Bad VCS log file *file-name*

Severity: Error

Explanation: VCS$IODL notifies that it has encountered a corrupted VCS log file.

User Action: None. VCS$IODL recovers from this error by renaming the corrupted file to VCS$19%%_%%_%%.LOG_BAD in the directory VCS$LOG, creates a fresh log file and proceeds.

BADLOGVER, Can't use bad version of file *file-name*

Severity: Warning

Explanation: Log file needed for REVIEW command has a bad version.

User Action: Cannot review period contained in the bad version of the log file.

BADPARAM, Bad communications message parameters

Severity: Error

Explanation: Internal software error possibly caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification of these images with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

BADPKT, Bad packet type field in received packet

Severity: Error

Explanation: Internal software error caused by an unexpected pipe-packet type. Therefore, the packet was ignored.

User Action: Depending on the packet that was ignored, recovery may not be necessary. If further problems become evident, restart the software.

BADPROC, Bad process identification code

Severity: Error

Explanation: Internal software error caused by some type of version skew between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the version skew with ANALYZE/IMAGE. Under image identification, check the image file identification. These fields should be the same for all VCS .EXE files. Reinstall the product or call software support.

BADRANGE, Only log files between Jan 1 1977 to Nov 9 2021 can be requested

Severity: Error

Explanation: When you issue the REVIEW command, you specified a date that is not within the accepted bounds.

User Action: Specify a date between Jan 1 1977 and Nov 9 2021.

BADSCANFIL, Scan profile *profile-name* for node *node-name* contains bad record

Severity: Warning

Explanation: One of the VCS 1.2 scan files for the specified node contains a record that is formatted incorrectly for data conversion.

User Action: Add the ill-formed record manually to the VCS 1.3 configuration using the new Configuration Editor.

BADSCANREC, Scan file format error, record = *scan-record*

Severity: Warning

Explanation: The VCS Data Converter tool displays the ill-formed scan record that cannot be converted.

User Action: Add the ill-formed record manually to the VCS 1.3 configuration using the new Configuration Editor.

BADTYPE, Record type unknown

Severity: Fatal

Explanation: The record type of the record does not match the types that the current routine can process.

User Action: Your configuration file my have become corrupted. Restore or rebuild the configuration file.

BADVER, Bad logfile version

Severity: Fatal

Explanation: During the course of initialization, the VCS$IODL found that today's log file already exists. It was found however, that the version written in the file and that known by the VCS$IODL were not the same. This indicates that the contents of the existing log file is unpredictable.

User Action: Shut down VCS. Move, rename, or delete the existing log file with today's date as a file name, and restart the VCS software.

BREAKYN, Do you want to transmit a break (Yes/No)?

Severity: Informational

Explanation: You have typed the break character to generate a Break signal on the console line to the serviced node. VCS types this message to confirm that request. If your answer is negative, the ASCII control character is forwarded to the node unchanged and the session continues. If your answer is positive, a Break signal is generated on the console line.

User Action: Reply with either a positive or negative response.

BUFFEROVF, Buffer overflow

Severity: Error

Explanation: Internal software error possibly caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification of these images with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

C3LOGIC, Internal logic error

Severity: Error

Explanation: An unexpected internal error was detected.

User Action: Take corrective action based on the accompanying message.

C3MEM, Virtual memory exhausted

Severity: Warning

Explanation: A memory allocation error occurred while loading the C^3 window labels. The labels will not be used.

User Action: Check user working set and page file quotas.

C3_DEF, Default icon placement will be used

Severity: Warning

Explanation: The icon placement information in the current VCS configuration file cannot be used to place the icons on the screen. Most often this will occur when a new node or peripheral was added to the configuration using the command line configuration editor. Since the command line configuration editor cannot used to choose or place icons, the new icons will have no placement

information. The default placement algorithm will place all the icons on the screen using calculated placement values. The display of the C³ window labels will also be suppressed.

User Action: Use the DECwindows configuration editor to correct the placement of any new icons. The editor will verify that no icon obscures another. Use the VCS RECONFIGURE command to correct the icon placement and restore the labels in the running C³ windows.

C3_DEF_ICON, Icon *icon-name* for *identifier* not found. A default icon has been assigned.

Severity: Warning

Explanation: The node or peripheral *identifier* is associated with the icon *icon-name* that was not found in the version of the VCS$DW_ICON_DFN.UID file that C³ is using.

User Action: Add missing icons to VCS$DW_ICON_DFN.UID using the icon conversion tool, then restart C³ Interface.

CDISABLE, System *node-name* has been disabled

Severity: Informational

Explanation: The VCS software has discontinued processing for the system *node-name*. This may be due to an error on the communication line or the result of the MONITOR DISABLE command.

User Action: If the failure is due to a LAT communication failure, attempt to reset the line by issuing the VCS RECONFIGURE command.

CENABLE, System *node-name* has been enabled

Severity: Informational

Explanation: The VCS software has restored services for the system *node-name*.

User Action: None.

CFGCLOS, Error closing configuration file

Severity: Error

Explanation: VCS$IODL completed processing the configuration file but found errors when trying to close it. The file may still be open.

User Action: No recovery is necessary. Note however, that you cannot edit this configuration file until you stop the VCS$IODL. Also, if the configuration file is open, the RECONFIGURE command may fail.

CFGCMDERR, *n* errors encountered, configuration file not updated

Severity: Warning

Explanation: Indicates the configuration file does not contain any of the changes made during the editing session. Changes were saved to the journal file as an unverified file.

User Action: To continue editing session invoke editor specifying the /INPUT and /OUTPUT qualifiers and correct the problems.

CFGERR, Error in configuration file

Severity: Warning

Explanation: VCS has discovered a syntax error in the configuration file. This check is done both at initialization and at reconfiguration before VCS tries to process data in the file.

User Action: At initialization time, a configuration file error stops VCS$IODL. At reconfiguration time, you receive only a warning. In either case, the configuration file may have been corrupted, use the Configuration Editor to rebuild it.

CFGNOCHANGE, No changes made to the configuration file

Severity: Informational

Explanation: You quit or exited from the Configuration Editor without saving any modifications.

User Action: None.

CFGOPN, Error opening configuration file

Severity: Error

Explanation: C^3 Interface was unable to open the current configuration file due to an error.

User Action: Take corrective action based on the accompanying message.

CFGOPN, Error opening configuration file

Severity: Error

Explanation: VCS$IODL found errors in the configuration file when trying to open it during an initialization or reconfiguration. With this error, a RECONFIGURE command will not be processed but the VCS$IODL will continue. If this error is encountered during initialization, this causes a fatal error.

User Action: Use the Configuration Editor to verify the validity of the configuration information.

CFGSKW, Configuration skew

Severity: Fatal

Explanation: You purged an older version of the configuration file which is still being used by the VCS software components. Subsequent activation of the VCS interactive interfaces or processes cannot locate that version of the configuration file.

User Action: Reconfigure VCS.

CFGUNVER, Configuration file is unverified

Severity: Warning

Explanation: The configuration file is not verified and cannot be used as a valid configuration file for VCS.

User Action: Use the verify command while in the editing session to list and fix the verification problems. Exiting from the editor verifies the file.

CFG_EXITCONT, Changes will not be saved - Continue exit? [No]

Severity: Informational

Explanation: Your configuration editor session cannot be terminated gracefully and changes that were made will not be saved in the configuration file. You are allowed to resume the editing session in order to correct any errors before trying to exit again.

User Action: Decide and response to the prompt.

CLSLOG, Error closing the VCS Console log file

Severity: Warning

Explanation: The C^3 Interface failed to close a console log file after having accessed it to retrieve data for the event context display. The exact nature of the failure is described in the following message.

User Action: Correct the problem as described by the following message.

CLUMAXSIZE, Cluster *name* has exceeded maximum size

Severity: Error

Explanation: Maximum cluster size is 32 nodes.

User Action: Remove nodes from the cluster definition to satisfy the limit requirement.

CLUNODEDUP, Cluster has duplicate member *node-name*

Severity: Error

Explanation: The specified node is already a member of the cluster definition.

User Action: None.

CLUNODENOF, Cluster *name* node *name* not found in configuration file

Severity: Warning

Explanation: One of the members of the cluster definition is not an entity in the configuration file.

User Action: Add the node to the configuration file.

CLUPORTDUP, Cluster *name* node *name* has duplicate port *number*

Severity: Error

Explanation: The port number assigned to the node in the cluster definition is already assigned to another cluster member.

User Action: Find the duplicate port and assign the correct port number.

CLUSTERCNT, Loaded *n* clusters

Severity: Informational

Explanation: The configuration data for *n* cluster records was successfully loaded.

User Action: None.

CMDCNT, Loaded *n* commands

Severity: Informational

Explanation: The configuration data for *n* command records was successfully loaded.

User Action: None.

CMDFAIL, Failed to execute command

Severity: Warning

Explanation: An error occurred executing a command from the Commands or pop-up menu.

User Action: Take corrective action based on the accompanying message.

COMPLETEREC, [VCS-completed line]

Severity: Informational

Explanation: The log data contains a complete line which has been recorded earlier as a partial line.

User Action: None.

CONFVRF, Configuration file failed to verify

Severity: Error

Explanation: The configuration file being used by VCS did not pass the necessary editor data verification tests. The C^3 interface will not run with this configuration file.

User Action: Use the Configuration Editor to verify the contents of the configuration file.

CONNECT, Connecting to system *node-name*

Severity: Informational

Explanation: The Monitor Interface notifies you of the system that you are connected to using the CONNECT command.

User Action: None.

CONVDATA, Processed *count* data file(s) and created: *statistics*

Severity: Informational

Explanation: The data conversion tool reports processing statistics at completion. The statistics include the number of node records, scan files, event records, and user records processed.

User Action: None.

CONVDONE, Conversion complete

Severity: Informational

Explanation: The conversion process completed successfully.

User Action: None.

CONVDUPEVT, Creating variant *number* for duplicated event *name*

Severity: Informational

Explanation: The conversion tool encounters an event having the same name as another previously converted one. It is creating a variation in the name of the latter, by appending a serial value *number* to the original event identifier *name*, to avoid duplication in the configuration file.

User Action: None.

CONVERTEDICON, Icon width and height in pixels *dimension* x *dimension*

Severity: Informational

Explanation: An icon was successfully converted by the Icon Converter with specified dimensions.

User Action: None.

CONVEVENTFAIL, Failed to convert event *name*

Severity: Warning

Explanation: The data conversion tool failed to convert and create a record in the new configuration file for the specified event. The exact nature of the failure is described in the following message.

User Action: Add the event manually to the VCS 1.3 configuration using the new Configuration Editor.

CONVEVENTREF, Scan profile *name* references event *event-name*

Severity: Warning

Explanation: The data conversion tool failed to convert the scan profile *name*. It notifies you the events that are referenced by that scan profile.

User Action: Add the scan profile manually to the VCS 1.3 configuration using the new Configuration Editor.

CONVEVTEXIST, Event *name* already exists as event *new-name*

Severity: Informational

Explanation: The data conversion tool encounters an event that it has already converted earlier and notifies you that it will not convert the same event again.

User Action: None.

CONVFAIL, Conversion failed

Severity: Fatal

Explanation: The data conversion tool failed to complete successfully.

User Action: Examine previous error messages, correct the errors and restart the conversion process.

CONVFILE, Converting *file-name*

Severity: Informational

Explanation: The specified file is being converted by the Data or Icon Converter.

User Action: None.

CONVFROM, Converting data from version *identifier* format

Severity: Informational

Explanation: The data conversion tool notifies you the previous version of VCS software which it is converting the configuration data from.

User Action: None.

CONVICON, Converting icon from file *file-name*

Severity: Informational

Explanation: The Icon Converter is in the process of converting the specified DDIF file.

User Action: None.

CONVLOGS, Converted log directory

Severity: Informational

Explanation: The data conversion tool notifies you that it has successfully converted the VCS log directory record.

User Action: None.

CONVNODE, Converted node *name*

Severity: Informational

Explanation: The data conversion tool notifies you that it has successfully converted the node *name* record.

User Action: None.

CONVNODEFAIL, Failed to convert node !AD

Severity: Warning

Explanation: The data conversion tool notifies you that it has failed to convert the node *name* record.

User Action: Examine the saved copy of the old VCS configuration file for any errors in the node *name* record and restart the conversion process.

CONVOVRDUP, Too many variants for duplicated event *name*

Severity: Warning

Explanation: The data conversion tool has encountered more than 1,000 events of the same name and cannot create any more variant records in the configuration file.

User Action: Specify different names as you add the remaining events manually to the VCS 1.3 configuration using the new Configuration Editor.

CONVPERIPH, Converted peripheral *name*

Severity: Informational

Explanation: The data conversion tool notifies you that it has successfully converted the peripheral *name* record.

User Action: None.

CONVPERIPHFAIL, Failed to convert peripheral *name*

Severity: Warning

Explanation: The data conversion tool notifies you that it has failed to convert the peripheral *name* record.

User Action: Examine the saved copy of the old VCS configuration file for any errors in the peripheral *name* record and restart the conversion process.

CONVSCAN, Converted scan profile *name* (*n* events, *n* scan profiles)

Severity: Informational

Explanation: The data conversion tool notifies you that it has successfully converted the scan profile *name* record which includes the specified number of events and subprofiles.

User Action: None.

CONVSCANEXIST, Scan *name* already exists

Severity: Informational

Explanation: The data conversion tool encounters a scan profile that it has already converted earlier and notifies you that it will not convert the same profile again.

User Action: None.

CONVSCANFAIL, Failed to convert scan profile *name* (*n* events, *n* scan profiles)

Severity: Warning

Explanation: The data conversion tool notifies you that it has failed to convert the scan profile *name* record.

User Action: Examine the saved copy of the old VCS configuration file for any errors in the scan profile *name* record and restart the conversion process.

CONVUSER, Converted user *name*

Severity: Informational

Explanation: The data conversion tool notifies you that it has successfully converted the user *name* record.

User Action: None.

CONVUSERFAIL, Failed to convert user *name*

Severity: Warning

Explanation: The data conversion tool notifies you that it has failed to convert the user *name* record.

User Action: Examine the system Rights List for possible errors and restart the conversion process.

CREATEFILE, Creating *file-name*

Severity: Informational

Explanation: The specified file is being created.

User Action: None.

CTYLOSS, Console data loss at *date/time*

Severity: Warning

Explanation: The user interface discovered that it was not keeping up with the volume of data from the remote node. As a result, some console data from VCS$IODL destined for this particular user interface have been lost.

User Action: VCS$IODL has already recorded the data in the VCS log file prior to making it available to the user interfaces. Thus, you can use the REVIEW command starting at the specified *date/time* to examine the data.

CUSTINIT, Error initializing C3 customize

Severity: Warning

Explanation: The initialization of the C^3 customize menus and boxes failed.

User Action: Take corrective action based on the accompanying message.

DATLOSS, Data loss in pipe (ring buffer overwrite)

Severity: Error

Explanation: While reading a pipe, the VCS$IODL discovered that it was not keeping up with the volume of data from the remote node. As a result, some console data destined for this interface have been lost. This data is not permanently lost. You can use the REVIEW command to examine the data.

User Action: If no other errors are evident, no recovery is necessary. If other errors are evident, restart the VCS$IODL.

DBLSIG, Encounter error while handling current error

Severity: Fatal

Explanation: The error handler detected an error within itself.

User Action: Restart the process.

DCLCOM, Incorrect /COMMAND specification

Severity: Error

Explanation: You issued a DCL VCS command with a qualifier. This message should accompany a DCL error message.

User Action: Correct the file specification and re-enter the command.

DCLOUT, Incorrect /OUTPUT specification

Severity: Error

Explanation: You issued a DCL VCS command with a qualifier. This message should accompany a DCL error message.

User Action: Correct the device specification and re-enter the command.

DCLSEC, Incorrect /SECTION specification

Severity: Error

Explanation: You issued a DCL VCS command with a qualifier. This message should accompany a DCL error message.

User Action: Correct the file specification and re-enter the command.

DECSERVER, The DECserver is not accessible

Severity: Error

Explanation: The DECserver, on which a LAT host initiated connect port was set up and mapped into the corresponding LT device, cannot be reached over the Ethernet.

User Action: Diagnose and correct the problem, then issue the VCS RECONFIGURE command to enable the failed communications path.

DECWINIT, Failure initializing the DECwindows interface

Severity: Error

Explanation: An error occurred in the process of initializing the DECwindows interface.

User Action: Take corrective action based on the accompanying message.

DEFAULTICONS, Some icons referenced in configuration file not found in DECW$SYSTEM_DEFAULTS:VCS$DW_ICON_DFN.UID, defaults assigned

Severity: Warning

Explanation: Icons are being referenced that no longer exist in DECW$SYSTEM_DEFAULTS:VCS$DW_ICON_DFN.UID.

User Action: Rebuild the DECW$SYSTEM_DEFAULTS:VCS$DW_ICON_DFN.UID file using the Icon Converter including the

missing icons; copy the newly created file to DECW$SYSTEM_DEFAULTS.

DEFBRK, Default break key set as *key-name*

Severity: Informational

Explanation: From this point, the default break key is set to *key-name* in the Monitor Interface. This is the default break key for the CONNECT and SELECT commands.

User Action: None.

DEFBRKCLR, Default break key cleared

Severity: Informational

Explanation: From this point, there is no default break key set in the Monitor Interface.

User Action: None.

DEFESC, Default escape key set as *key-name*

Severity: Informational

Explanation: From this point, the default escape key is set to *key-name* in the Monitor Interface. This is the default escape key for the CONNECT, SELECT and CYCLE commands.

User Action: None.

DEFFONT, Using default font

Severity: Warning

Explanation: The requested font could not be loaded by the server, the default font of the server will be used.

User Action: The server may be low on resources try reducing number of applications running on the server at the same time. Also check to make sure the specific font exists on that server.

DELETEALL, *count* deleted from configuration file

Severity: Success

Explanation: The Configuration Editor successfully deleted the specified number of records from the configuration file as it performed a wildcard delete request.

User Action: None.

DELETED, *identifier* deleted from configuration file

Severity: Success

Explanation: The Configuration Editor successfully deleted the specified record from the configuration file.

User Action: None.

DEVALLOC, Device *device-name* for system *node-name* cannot be allocated

Severity: Warning

Explanation: VCS$IODL cannot allocated the device *device-name* for communication with the system *node-name*.

User Action: Run the Configuration Editor to verify the existence of the device *device-name*. If the device does exist on the system, verify that it is not currently being allocated by another process.

DISABLE, System is disabled

Severity: Informational

Explanation: Communication with this system has temporarily been discontinued by the System Manager with the Monitor Interface DISABLE command.

User Action: None.

DISPLAYONLY, Illegal command for a display only session

Severity: Warning

Explanation: The user entered the SELECT, OUTPUT, CONNECT, DISABLE, or ENABLE command in a display-only Monitor Interface session.

User Action: Exit the current Monitor Interface session and start another one using the /ACCESS_MODES=INTERACTIVE qualifier before issuing one of the above commands.

DPYINIT, Can't initialize the DECwindows display *display-name*

Severity: Error

Explanation: C^3 was unable to start a graphical display on the node, server and screen specified by *display-name*.

User Action: Check the session manager security entries for the logged in user on display *display-name*.

DUPCFGDEV, Duplicate device name in configuration: *device-name*

Severity: Warning

Explanation: Two systems listed in the configuration file have the same device name. This is a fatal error during initialization and a warning during reconfiguration.

User Action: Edit the configuration file and insure that each system listed has a unique terminal device associated with it.

DUPICON, *icon name* icon already exists in output file, no conversion

Severity: Error

Explanation: No conversion was performed on the specified icon; An icon with the same name has already been converted and exists in the output file.

User Action: Change the VMS file name of the DDIF source file for the icon.

DUPLICATE, *Configuration-entity* already entered in configuration file

Severity: Error

Explanation: Duplicate records are not allowed in the configuration file.

User Action: Re-enter the specified entity with a different name.

DUPMBX, Duplicate mailbox

Severity: Error

Explanation: During initialization, when VCS$IODL tried to create its mailbox, it discovered the system already had a mailbox by that name. The usual cause is another VCS$IODL already running on the system.

User Action: Find the other VCS$IODL and stop it before trying to run another, or use VCS$STARTUP.COM which automatically checks and stops another VCS$IODL.

DUPMBX, Duplicate mailbox

Severity: Error

Explanation: During initialization, when the C³ process tried create its mailbox, it discovered the system already had a mailbox by that name. This could be due to a second C³ process running in the same job or a problem creating a unique mailbox name for a C³ subprocess.

User Action: Create another detached job to run the second C³ display. If the problem appeared when starting a C³ subprocess, stop and restart any C³ displays.

DUPPRCNAM, Attempt to create a duplicate process

Severity: Warning

Explanation: The user is running the Central Control Coordinator application and has attempted to re-spawn an existing task.

User Action: The name of the duplicate process appears in the message window with the warning. Execute a SHOW SYSTEM and look for the process name. Either find a way to attach to it and execute it, or issue a STOP/ID command and terminate it.

DUPSYS, Duplicate system name *node-name* ignored

Severity: Warning

Explanation: You have specified the same system more than once in the VCS command line.

User Action: Enter the list of system names again without duplication.

DW_CB_REG_FAIL, Cannot register fields from UID file *file-name*

Severity: Error

Explanation: The registration of the fields from the specified filename has failed.

User Action: Run IVP make sure all VCS .UID files exist and are in their correct locations. Try restarting DECwindows and VCS interface. If the problem still exists, contact software support.

DW_CLIP, Cannot convert clip mask, Using scanning conversion

Severity: Warning

Explanation: The foreground and background colors of the icon that must change color for the current event prevent the clip mask version of the icon from being computed. The icon may be corrupted when events are processed.

User Action: Change the icon to one of the default set provided in the kit. Modify and rebuild any user supplied icons.

DW_DEF_ICON, Unable to find icon *icon-name*, a default icon has been assigned

Severity: Informational

Explanation: The icon being referenced no longer exists in DECW$SYSTEM_DEFAULTS:VCS$DW_ICON_DFN.UID.

User Action: Quit the interface, rebuild the VCS$DW_ICON_ DFN.UID file using the Icon Converter including the missing icon; copy the newly created file to DECW$SYSTEM_DEFAULTS.

DW_DEL_CLUSTER, Confirm deletion, Cluster: *cluster-name*

Severity: Informational

Explanation: Confirm deletion of cluster definition.

User Action: None.

DW_DEL_EVENT, Confirm deletion, Event: *event-name*

Severity: Informational

Explanation: Confirm deletion of event definition.

User Action: None.

DW_DEL_LABEL, Confirm deletion, Label: *label-name*

Severity: Informational

Explanation: Confirm deletion of label definition.

User Action: None.

DW_DEL_NODE, Confirm deletion, Node: *node-name*

Severity: Informational

Explanation: Confirm deletion of node definition.

User Action: None.

DW_DEL_PERIPH, Confirm deletion, Peripheral: *peripheral-name*

Severity: Informational

Explanation: Confirm deletion of peripheral definition.

User Action: None.

DW_DEL_SCAN, Confirm deletion, Scan Profile: *scan-name*

Severity: Informational

Explanation: Confirm deletion of scan profile definition.

User Action: None.

DW_DEL_USER, Confirm deletion, User: *user-name*

Severity: Informational

Explanation: Confirm deletion of user definition.

User Action: None.

DW_DISPEXE, Maximum of *n* C3 Displays allowed, entry ignored

Severity: Error

Explanation: The maximun number of C^3 display windows has been reached.

User Action: Remove some C^3 displays from the list.

DW_DWT_ERR, DECwindows Toolkit Error: *message-text*

Severity: Error

Explanation: Error message from DECwindows toolkit.

User Action: Check DECwindows documentation for possible causes of the specified problem; take action to correct the problem. If this does not help contact software support.

DW_DWT_WARN, DECwindows Toolkit Warning: *message-text*

Severity: Warning

Explanation: Warning message from DECwindows toolkit.

User Action: Check DECwindows documentation for possible causes of the specified problem; take action to correct the problem. If this does not help contact software support.

DW_FETCH_FAIL, Cannot fetch widget *name* from UID file *file-name*

Severity: Error

Explanation: The fetching of a DECwindows toolkit widget from the specified filename has failed.

User Action: Run IVP make sure all VCS UID files exist and are in their correct locations. Try restarting DECwindows and VCS, if the problem still exists contact software support.

DW_INSFMEM, Insufficient memory to support internal data structures. Restarting the interface may help.

Severity: Error

Explanation: The Icon Converter has run out of virtual memory.

User Action: Try restarting interface.

DW_JNL_SAVE, Saving modifications to journal file *file-name*

Severity: Informational

Explanation: The verification of the configuration file has failed and changes were saved to the journal file specified.

User Action: Start the editor using the /input and the /output qualifier to continue editing session of the unverified file.

DW_LIT, Can't fetch UID literal *literal*

Severity: Warning

Explanation: An error occurred accessing the VCS$C3.UID file to get the string literal named *literal*.

User Action: Quit and restart the application. Reinstall the VCS software kit.

DW_MRS_EXCEEDED, Maximum record size exceeded

Severity: Error

Explanation: Configuration-entity record size has exceeded maximum size.

User Action: Remove some items from record definition.

DW_NEW_ICON, New icon added to configuration window

Severity: Informational

Explanation: A new icon has been added to the middle of the configuration window.

User Action: None.

DW_NEW_LABEL, New label added to configuration window

Severity: Informational

Explanation: A new label has been added to the middle of the configuration window.

User Action: None.

DW_NODENOTVIS, Icon for node *node-name* is not visible.

Severity: Error

Explanation: DECwindows verification for the configuration file has failed because a node icon has been occluded.

User Action: Find hidden icon and make it visible on the screen.

DW_NOLABEL, Unable to find label *name* in configuration.

Severity: Error

Explanation: Specified label was not found in configuration file.

User Action: Check label case, labels are case sensitive.

DW_NO_ICONS, Cannot find icon definitions in UID file *file-name*

Severity: Error

Explanation: No icons definitions were found in the specified UID file.

User Action: Create icon UID file using icon converter. Move the newly created UID file to DECW$SYSTEM_DEFAULTS:.

DW_NO_VCS_ICON, Unable to fetch VCS icon.

Severity: Error

Explanation: Attempt to fetch VCS icon failed.

User Action: Run IVP to make sure all VCS UID files exist and are in their correct locations. Try restarting DECwindows and VCS interface, if the problem still exists contact software support.

DW_NOPRODICON, Unable to fetch product icon

Severity: Error

Explanation: VCS product icon was not created.

User Action: Use IVP to verify all VCS files are in there correct places; try restarting DECwindows interface, if this does not correct the problem contact software support.

DW_NOSHELL, Unable to create application shell

Severity: Error

Explanation: The DECwindows application shell was not created.

User Action: Try restarting DECwindows and VCS interface, if this does not correct the problem contact software support.

DW_PERIPHNOTVIS, Icon for peripheral *name* is not visible

Severity: Error

Explanation: DECwindows verification for the configuration file has failed because a peripheral icon was occluded.

User Action: Find the hidden icon and make it visible on the screen.

DW_PIXMAP, Can't get default pixmap for *identifier*

Severity: Warning

Explanation: There was an error in processing the icon for node or peripheral *identifier*. This may be due to a problem with the DECwindows server you are using or with the icon that is used by the node or peripheral *identifier*.

User Action: Try C^3 on another server. Change the icon to one of the default set provided in the kit. Rebuild any user supplied icons.

DW_REPLCLUS, Replace cluster *name* with cluster *name*

Severity: Informational

Explanation: The cluster name field has been changed while modifying a cluster definition.

User Action: Respond with yes to change the name of the cluster definition; no to make a copy of the cluster definition with a new name.

DW_REPLEVENT, Replace event *name* with event *name*

Severity: Informational

Explanation: The event name field has been changed while modifying a event definition.

User Action: Respond with yes to change the name of the event definition; no to make a copy of the event definition with a new name.

DW_REPLNODE, Replace node *node* with node *name*

Severity: Informational

Explanation: The node name field has been changed while modifying a node definition.

User Action: Respond with yes to change the name of the node definition; no to make a copy of the node definition with a new name.

DW_REPLPER, Replace peripheral *name* with peripheral *name*

Severity: Informational

Explanation: The peripheral name field has been changed while modifying a peripheral definition.

User Action: Respond with yes to change the name of the peripheral definition; no to make a copy of the peripheral definition with a new name.

DW_REPLSCAN, Replace scan profile *name* with scan profile *name*

Severity: Informational

Explanation: The scan profile name field has been changed while modifying a scan profile definition.

User Action: Respond with yes to change the name of the scan profile definition; no to make a copy of the scan profile definition with a new name.

DW_REPLUSER, Replace user *name* with user *name*

Severity: Informational

Explanation: The user name field has been changed while modifying a user definition.

User Action: Respond with yes to change the name of the user definition; no to make a copy of the user definition with a new name.

DW_SYNC_ERR, Displayed and internal data structures are out of sync. Restart the editor to reset structures

Severity: Error

Explanation: Internal software error, internal data structures and DECwindows display do not correspond.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message. If no other error are apparnet, restart the failed component.

DW_TITLE, Can't fetch window title *title-name*

Severity: Warning

Explanation: An error occurred accessing the VCS$C3.UID file to get the window title named *title-name*.

User Action: Quit and restart the application. Reinstall the VCS software kit.

DW_UIDCTX, Can't get UID resource context

Severity: Warning

Explanation: A UID resource context record could not be allocated.

User Action: Quit and restart the application. Reinstall the VCS software kit.

DW_UID_FILE_ERR, Cannot load UID files *file-name*. Check SYS$LIBRARY area.

Severity: Error

Explanation: An error occurred accessing the VCS$C3.UID file to get the window title named *title name*.

User Action: Quit and restart the application. Reinstall the VCS software kit.

DW_VCSNOTVIS, VCS icon is not visible

Severity: Error

Explanation: DECwindows verification for the configuration file has failed because the VCS icon has been occluded.

User Action: Find the hidden VCS icon and make it visible on the screen.

DW_XLIB_ERR, Error during XLIB routines: ...

Severity: Error

Explanation: X library error message.

User Action: Check DECwindows documentation for possible causes of the specified problem; take action to correct the problem, if this does not help contact software support.

DW_XRMDB, Can't get X resource database

Severity: Warning

Explanation: The X windows resource database pointer could not be read.

User Action: Restart the X windows server software. Try the application on another server.

ENABLE, System is enabled

Severity: Informational

Explanation: Communication with this system has been restored by the System Manager with the Monitor Interface ENABLE command.

User Action: None.

ERRHAND, Failed to register error handler

Severity: Warning

Explanation: C^3 was unable to set up the routines to be used when the image is stopped abnormally.

User Action: Take corrective action based on the accompanying message.

ERROR_DISABLE, System *node-name* has been permanently disabled

Severity: Warning

Explanation: The communications path between the VCS and the console port of *node-name* has failed. Access to *node-name* has been disabled.

User Action: Diagnose and correct the problem, then issue the VCS RECONFIGURE command to enable the failed communications path.

EVCOUNTINV, Invalid line count for event *name*

Severity: Error

Explanation: The context count for an event definition is invalid.

User Action: Changed event definition so contex count falls in the range 0-1023.

EVENTCNT, Loaded *n* events

Severity: Informational

Explanation: The configuration data for *n* event records was successfully loaded.

User Action: None.

EVPRIORINV, Invalid event priority for event *name*

Severity: Error

Explanation: Event priorities can only be in the range of 0-100.

User Action: Re-enter with correct priority.

EVSTARTINV, Invalid start value for event *name*

Severity: Error

Explanation: VCS only allows values in the range of 0 to 63.

User Action: Re-enter using correct value.

EXITPRC, Exit from an existing session or increase maximum window setting

Severity: Warning

Explanation: No additional subprocesses can be spawned through the Central Control Coordinator.

User Action: An existing spawned process should be terminated (for example, exit a Monitor Interface session), or increase the subprocess limit by changing the terminal windows limit in the C^3 general customize dialog box.

EXITYN, Do you want to exit (Yes/No)?

Severity: Informational

Explanation: You have typed the escape character to disconnect from the current Connect session. VCS types out this message to confirm that request. If your answer is negative, the ASCII control character is forwarded to the node unchanged and the session continues. If your answer is positive, control is returned to the VCS system.

User Action: Reply with either a positive or negative answer.

EXMAXSCNREC, Scan profile *profile-name* for node *node-name* contains *n* records

Severity: Warning

Explanation: The scan profile contains more than the maximum allowed (128) number of events.

User Action: Edit the appropriate scan profile and reduce the total number of events to 128 or less.

EXSCNIGNORE, Only 128 highest priority records are used for scanning

Severity: Warning

Explanation: Not all scan strings contained in the scan profile will be detected because your scan profile exceeded the maximum entries allowed (128).

User Action: Edit the appropriate scan profile and reduce the total number of events to 128 or less.

FAILINK, Failed to link to node *node-name*

Severity: Error

Explanation: Failed to connect to a pipe for a system. This is an internal software error.

User Action: Restart the Monitor Interface. If it fails again, restart VCS.

FILENOTCREATED, Cannot create *file-name*

Severity: Error

Explanation: The specified file was not created due to a previus error.

User Action: Examine the previous error and determine what caused the file create to fail; take appropriate action to correct the problem.

FILENOTFOUND, Cannot find *file-name*

Severity: Error

Explanation: The specified file was not found.

User Action: Check the file specification and protection.

FILENOTOPEN, Cannot access *file-name*

Severity: Error

Explanation: The specified file was not opened due to previous error.

User Action: Examine the previous error and determine what caused the file open to fail; take appropriate action to correct the problem.

FILENOTUIL, *file-name* is not in icon conversion file format, no conversion

Severity: Error

Explanation: Icon converter was unable to append to existing file of unrecognized format.

User Action: Make sure the file being appended to was created by the Icon Conversion Tool or recreate all icons from their DDIF source files.

FILNOTOPN, File *file-name* not opened due to previous error

Severity: Error

Explanation: The named file cannot be opened due to a previous error.

User Action: Examine the previous error and determine what caused the file open to fail; take appropriate action to correct the problem.

FLSETERM, Failed to setup user terminal

Severity: Error

Explanation: The Monitor Interface failed to set up the correct characteristics for a user terminal. This may indicate a line problem.

User Action: Restart the Monitor Interface.

FULLDELETE, Disk full action taken, log file *file-name* deleted

Severity: Informational

Explanation: The number of free blocks on the logging disk has dropped below the disk-full-delete percentage (normally 1%) of the total size of the disk or 1000 blocks, whichever is larger. To make space, the VCS$IODL has deleted the oldest VCS log file (any file other than today's is old). The name of the deleted file is displayed in the message. VCS$IODL will continue to delete files periodically maintaining sufficient space on the disk.

User Action: If you want to keep the old files, you must copy them periodically from this disk to another storage medium. The number of free blocks must be greater than the disk-full-delete percentage of the total size of the disk.

FULLDELIF, Disk full action taken, log file deleted

Severity: Informational

Explanation: VCS$IODL determined that disk space is getting low on the logging disk. VCS$IODL deleted an old VCS log file to free disk space for new data.

User Action: If you want to save log files, copy them to another storage device, and then delete them from this disk.

FULLELOGERR, Disk full action failure, error deleting old event file

Severity: Error

Explanation: VCS$IODL process encountered an error when trying to delete old Event Log files to free up space on the disk. This error prevented the deletion of files.

User Action: Delete files from the logging disk to make the space that VCS$IODL was trying to obtain.

FULLERR, Disk full action failure, error deleting old log file

Severity: Error

Explanation: VCS$IODL encountered an error when trying to delete old log files to free up space on the disk. This error prevented the deletion of the files.

User Action: Delete files from the logging disk to make the space the VCS$IODL was trying to obtain.

FULLNOFILE, No log file to delete

Severity: Error

Explanation: VCS$IODL found the logging disk does not have enough free space, even after having deleted all old log files to make room for the new data.

User Action: Delete additional unused files to make room for new log data.

FULLWARN, Disk full warning, please free space on logging disk

Severity: Warning

Explanation: During a periodic check of free space, VCS$IODL determined that the number of free blocks has dropped below the disk-full-warning percentage (normally 2%) of the total size (in blocks) of the disk or 2000 blocks, whichever is larger. VCS$IODL will not delete any files at this time; this is just a warning. The log file archival procedure (SYS$MANAGER:VCS$ARCHIVE.COM) will also be activated at this time if it exists.

User Action: Delete enough files to get the number of free blocks back above the disk-full-warning percentage of the total size (in blocks) of the disk.

HALFDUPLEX, Half-duplex connection

Severity: Informational

Explanation: When the Connect Interface is initiated from a batch job with the /INPUT qualifier, it indicates that no output from the system will be displayed at the output device.

User Action: None.

ICONKIND, ICON *icon-number* does not exist

Severity: Warning

Explanation: The color of an icon cannot be changed because the icon was not processed correctly when C³ was started. This may be due to a problem with the DECwindows server you are using or with the icon that is used by the node or peripheral. There may be a matching VCS-W-DW_PIXMAP error for this icon when C³ was started.

User Action: Try C³ on on other server. Change the icon to one of the default set provided in the kit. Rebuild any user supplied icons.

ILLMSG, Illegal message

Severity: Error

Explanation: The requested message number could not be found in the VCS Error file.

User Action: Reinstall the VCS software kit.

IMGCOMP, Invalid icon *icon name*, compressed images unsupported in icon conversion

Severity: Error

Explanation: The specified icon was not converted because the DDIF source file was in compressed image format.

User Action: Create icons with graphics editor that saves images in uncompressed format.

INIT, *component* initialization complete at *date/time*

Severity: Informational

Explanation: *Component* is either the VCS$IODL, VCS$SCANNER, C³ or the VCS Recorder. This message is sent after the VCS$IODL, C³, the Scanner or the Recorder has completed the initialization process.

User Action: None.

INITWITHERR, *process* initialization completed with errors at *date/time*

Severity: Error

Explanation: The named process initialized with errors.

User Action: Examine the information in the log file for the named process and determine appropriate action based on that information.

INSFPRIV, Insufficient privileges

Severity: Error

Explanation: You are attempting to execute a VCS image without authorization.

User Action: See the system manager to obtain proper privileges.

INSFRES, Insufficient resources

Severity: Error

Explanation: There are insufficient resources to perform the current function.

User Action: See the system manager to obtain proper user account quotas.

INTERRUPTED, The terminal line path has been interrupted

Severity: Error

Explanation: The communications path between the VCS and a console port has failed.

User Action: Diagnose and correct the problem, then issue the VCS RECONFIGURE command to enable the failed communications path.

INVAPPL, Error with user application file, *file-name*

Severity: Warning

Explanation: The application file cannot be loaded by the Access Interface.

User Action: Verify the application has been compiled and linked properly and that the logical name translates correctly.

INVDELPCNT, Disk full warning percent must be greater than delete percent

Severity: Error

Explanation: Disk full warning percentage cannot be less than disk full delete percentage.

User Action: Re-enter with warning percent less than delete percent.

INVFUNC, Invalid logging function code

Severity: Fatal

Explanation: An internal request for logging contained an invalid function code. This brings into question the integrity of the log file and the internal queue of logging requests. This is an internal software error.

User Action: Restart the VCS$IODL.

INVKEY, Invalid Key *name*

Severity: Error

Explanation: The record key has an invalid character. The key is typically the name of the entity being added, modified or deleted from the configuration file.

User Action: Re-enter the name with valid characters in the key.

INVPERCENT, Invalid percentage, specify 1 to 100

Severity: Error

Explanation: Invalid warning or delete percentage.

User Action: Specifiy a value from 1 to 100.

INVSYS, Invalid node *node-name* specified in command

Severity: Warning

Explanation: The argument did not match any node name or logical name known to VCS.

User Action: You probably made a typing error. If so, retype the command string. If not, check the node names with SHOW NODES, or if you used a logical name to represent a node name, execute the SHOW LOGICAL command.

LABELCNT, Loaded *n* labels

Severity: Informational

Explanation: The configuration data for *n* label records was successfully loaded.

User Action: None.

LATDISCON, LAT device *device-name* for system *node-name* has been disconnected

Severity: Warning

Explanation: The communications path over the Ethernet between the VCS device *device-name* and the console port of *node-name* has failed.

User Action: Diagnose and correct the problem, then issue the VCS RECONFIGURE command to enable the failed communications path.

LATRECONN, LAT device *device-name* for system *node-name* has been reconnected

Severity: Informational

Explanation: The failed communications path over the Ethernet between the VCS device *device-name* and the console port of *node-name* has been restored.

User Action: None.

LDLABEL, Error in main window labels

Severity: Warning

Explanation: An error occurred displaying the main window labels defined in the configuration file.

User Action: Take corrective action based on the accompanying message.

LINEDIS, Line is now disabled

Severity: Warning

Explanation: The system you targeted for output is disabled.

User Action: ENABLE the system.

LINEDISERR, Attempt to disable a disabled line

Severity: Warning

Explanation: You tried to disable a system that is already disabled.

User Action: None.

LINEERR, Error on the terminal line to system *node-name*

Severity: Warning

Explanation: Either the hardware line supporting the system is broken or the line characteristics are incorrect, for example, type ahead characteristics.

User Action: Repair the line or set the line characteristics.

LINENERR, Attempt to enable an enabled line

Severity: Warning

Explanation: You tried to enable a system that is already enabled.

User Action: None.

LINESOUT, *n* lines have been output; *n* characters received

Severity: Informational

Explanation: You just issued the REVIEW command. The first number is the number of lines the REVIEW command collected based on the supplied arguments. The second number is the actual number of characters collected from the specified console line(s).

User Action: This information tells you how much data you just received.

LINESTAT, Attempt to disable a disabled line, or enable an enabled line.

Severity: Error

Explanation: The node you tried to enable or disable is already enabled or disabled.

User Action: None.

LINESWAP, Terminal line assignment has changed

Severity: Warning

Explanation: After a reconfiguration, a known node has been moved to a different terminal line on the VCS system.

User Action: None.

LOCKOFF, Cannot unlock this node

Severity: Error

Explanation: You tried to use a SET LOCK OFF command; however, you do not own this lock. You cannot unlock it. The lock can be released only by the process setting the lock.

User Action: The SHOW NODES command indicates the owner of the lock. To unlock the node, the owner can release the lock by exiting or by using the SET LOCK OFF command.

LOCKON, Cannot lock this system

Severity: Error

Explanation: This system is already locked.

User Action: The SHOW NODES command indicates the owner of the lock. To unlock the node, the owner can release the lock by exiting or by using the SET LOCK OFF command.

LOCKOUT, Cannot send data to system locked by another user

Severity: Error

Explanation: The targeted console line is locked by another user. Therefore, you cannot send characters to it.

User Action: The SHOW NODES command indicates the owner of the lock. To unlock the node, the owner can release the lock exiting or by using the SET LOCK OFF command.

ILLOPT, Illegal option

Severity: Informational

Explanation: You specified an unrecognized option in the VCS command line.

User Action: Check the documentation for correct information and enter the command again.

LOGDIRDUP, Logging directory list has duplicate *name*

Severity: Error

Explanation: The logging directory list has a duplicate name.

User Action: Re-enter without duplicate.

LOGDIRNOTFOU, Logging directory *name* not found

Severity: Error

Explanation: The logging directory does not exist.

User Action: Re-enter with correct directory specification.

LONGCSTR, Too many DECterm customization options or titles too long

Severity: Error

Explanation: The list of customization options for the new DECterm window is longer than the allocated buffer.

User Action: Quit and restart the C^3 display. Submit a software performance report (SPR) with a copy of your VCS$CONFIGURATION.DAT file if restarting C^3 or VCS does not correct the problem.

MAXICONEXCEEDED, Invalid icon *icon name*, maximum icon size exceeded, maximum size allowed 250 X 250 pixels

Severity: Error

Explanation: The icon was not converted because it has exceeded the maximum icon size.

User Action: Use graphics editor to reduce dimensions of the icon.

MAXNAMELEN, Maximum icon name length exceeded for *icon name*, truncated to first 18 characters

Severity: Warning

Explanation: The Icon Converter truncated the name of the DDIF file to use for the corresponding icon name.

User Action: If you dont want the name truncated reduce the DDIF filename to 18 characters or less.

MAXSCANCNT, Maximum subprofile and event count has been exceeded for scan profile *name*

Severity: Error

Explanation: The combined scan subprofile and event count cannot exceed 34 for each scan profile definition.

User Action: Re-enter with correct limits.

MODIFIED, *identifier* modified

Severity: Success

Explanation: Confirmation that the specified configuration file entity has been modified.

User Action: None.

MONOCHROME, Server cannot support color icons. Using monochrome patterns.

Severity: Warning

Explanation: The DECwindows server that the C^3 is running on does not have the resources to allow the icons to change color to red, yellow and green in response to scan events. When an event occurs, the icons will be filled with a pattern that represents the priority of the scan events. This error is triggered when the server visual class is GrayScale or StaticGray or when the server visual class is PseudoColor and there are not enough free color table entries.

User Action: If the server class is PseudoColor, free up color table entries by quitting or changing the colors on other concurrent applications that make use of color.

MONS_AFTSHUT, Invalid function after VCS is no longer running

Severity: Warning

Explanation: You tried CONNECT, OUTPUT, RECONFIGURE, SELECT or SHUTDOWN after VCS was already shut down.

User Action: Restart VCS before using the function.

MONS_BUFFERR, Error in update buffer at line *line_number*

Severity: Warning

Explanation: This is an internal software error in the Monitor Interface TPU code.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message.

MONS_CMDERR, Error in process command line at line *line_number*

Severity: Warning

Explanation: This is an internal software error in the Monitor Interface TPU code.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message.

MONS_CMDINV, Invalid to provide partial commands in command file

Severity: Warning

Explanation: You have included a partially formed Monitor Interface command in an indirect command file and executed that command file with the @ command.

User Action: Edit the indirect command file to complete or remove the partially formed command.

MONS_CONDISA, Cannot connect *node-name*, system is disabled

Severity: Warning

Explanation: The node to which you tried to connect is disabled; therefore, you cannot connect to it.

User Action: You must issue the ENABLE system command before you can connect to it.

MONS_CONLOCK, Cannot connect *node-name*, locked by user *user-name*

Severity: Warning

Explanation: You tried to connect to a locked system.

User Action: Request the user who owns the lock to SET LOCK OFF.

MONS_CYCLERR, CYCLE command is ignored when issued from an indirect command file

Severity: Warning

Explanation: You have included the CYCLE command in an indirect command file and executed that command file with the @ command. The CYCLE was ignored by the Monitor Interface.

User Action: Remove the CYCLE command from the indirect command file.

MONS_ERROR, Error in call user at line *line_number*

Severity: Warning

Explanation: This is an internal software error in the Monitor Interface TPU code.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message.

MONS_FAILCB, Failed to create buffers necessary for Monitor Interface

Severity: Warning

Explanation: This is an internal software error in the Monitor Interface TPU code.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message.

MONS_FAILCW, Failed to create windows necessary for Monitor Interface

Severity: Warning

Explanation: This is an internal software error in the Monitor Interface TPU code.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message.

MONS_FAILLOCK, Can only change lock for a system

Severity: Warning

Explanation: The current log window does not contain system log information.

User Action: VIEW the desired system then issue the SET LOCK command.

MONS_INVSEL, Invalid selection

Severity: Warning

Explanation: You entered the VIEW command to display a non-existent Monitor Interface buffer.

User Action: Enter the VIEW command again specifying a valid buffer.

MONS_INVSYS, Invalid system specified, *node-name*

Severity: Warning

Explanation: You entered the CONNECT command from the Monitor Interface to communicate with system *node-name* which is not in the list of currently monitored nodes.

User Action: Enter the CONNECT command again specifying one of the nodes which are currently being monitored. If the node you wish to communicate with is not one of those, exit from the Monitor Interface and execute the VCS CONNECT command from DCL.

MONS_NEGSHIFT, Cannot shift by a negative amount

Severity: Warning

Explanation: You entered the SHIFT command to shift the display beyond the margins.

User Action: Enter the SHIFT command again with the proper amount of columns to shift the display.

MONS_NEXTEVENT, NEXT EVENT is valid only in the event log or event context window

Severity: Informational

Explanation: You entered the NEXT EVENT command while not viewing the event log or context log buffer.

User Action: Enter the VIEW/EVENT or VIEW/CONTEXT command before issuing the NEXT EVENT again.

MONS_NOCTXT, No context has been specified for this event

Severity: Informational

Explanation: The current event that you are viewing does not have any context information associated with it.

User Action: None.

MONS_NODEFAULT, No system name available for review default

Severity: Warning

Explanation: You entered the REVIEW command without a node name parameter while the current window is not displaying a node's console log.

User Action: Enter the REVIEW command again specifying the node for which the console data is to be retrieved.

MONS_NOEVENT, No current event is available

Severity: Informational

Explanation: You entered the VIEW/CONTEXT command and the Monitor Interface has not detected any events since it was started.

User Action: None.

MONS_NOEVENTS, No further events are available at this time

Severity: Informational

Explanation: You entered the NEXT EVENT command and the current event is the latest one detected by the Monitor Interface.

User Action: None.

MONS_NOKEYDEF, Key not previously defined

Severity: Informational

Explanation: You entered the UNDEFINE/KEY and specified a key that had not been defined.

User Action: None.

MONS_NOLSHIFT, Cannot shift past left margin

Severity: Warning

Explanation: You entered the SHIFT LEFT command and specified the number of columns beyond the left margin.

User Action: Enter the SHOW MONITOR to determine the number of columns you can specify in the SHIFT LEFT command.

MONS_NOPREVIOUS, No previous events available

Severity: Informational

Explanation: You entered the PREVIOUS EVENT command and the current event is the earliest one still maintained by the Monitor Interface in the event log buffer.

User Action: None.

MONS_NOSEARCH, No string to search

Severity: Informational

Explanation: You entered the FIND command without specifying a text string to be searched for.

User Action: Enter the FIND command again specifying a text string to be searched for.

MONS_ONEWDW, Already in one window mode

Severity: Informational

Explanation: You entered the ONE_WINDOW command while the Monitor Interface has already been displaying one log window on the terminal screen.

User Action: None.

MONS_ONLYONE, Only one window

Severity: Informational

Explanation: You entered the OTHER_WINDOW command while the Monitor Interface has been displaying only one log window on the terminal screen.

User Action: None.

MONS_PREVEVENT, PREVIOUS EVENT is valid only in the event log or event context window

Severity: Warning

Explanation: You entered the PREVIOUS EVENT command while not viewing the event log or context log buffer.

User Action: Enter the VIEW/EVENT or VIEW/CONTEXT command before issuing the PREVIOUS EVENT again.

MONS_REVIEWDIR, Review/Directory: This command is obsolete - use SHOW DIRECTORY instead

Severity: Informational

Explanation: You entered the REVIEW/DIRECTORY command which is an obsolete command.

User Action: Use the SHOW DIRECTORY command instead.

MONS_REVIEWERR, Invalid review argument for system name

Severity: Warning

Explanation: This is an internal software error in the Monitor Interface code.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message.

MONS_SEARCHFAIL, Failed to find string *string*

Severity: Informational

Explanation: You entered the FIND command and the specified text string *string* does not exist in the current window.

User Action: None.

MONS_SELDISA, Cannot select *node-name*, system is disabled

Severity: Warning

Explanation: The command failed because you cannot select a disabled node.

User Action: You must issue the ENABLE system command before you can SELECT it.

MONS_SELLOCK, Cannot select *node-name*, locked by user *user-name*

Severity: Warning

Explanation: You cannot SELECT a locked system.

User Action: Request the user who owns the lock to SET LOCK OFF.

MONS_SELSYS, The current window does not display a VCS node

Severity: Warning

Explanation: You are viewing a Monitor Interface window that is not a system log window, and you entered the SELECT or CONNECT command without specifying the system name.

User Action: Enter the SELECT or CONNECT command again with a system name.

MONS_SETWIDTH, Width set to *size*

Severity: Informational

Explanation: You entered the SET WIDTH command and the Monitor Interface confirmed the requested function.

User Action: None.

MONS_SHIFTLEFT, String was found to left of display - use SHIFT LEFT to see it

Severity: Informational

Explanation: You entered the FIND command and the specified text string was found on the left of the display window. This was because of one or more SHIFT RIGHT commands had been entered prior to the FIND command.

User Action: Enter the SHIFT LEFT command to view the found information.

MONS_SHIFTMARGN, Could not shift by the specified amount - shifted to left margin instead

Severity: Informational

Explanation: You entered the SHIFT LEFT command and specified the number of columns to be shifted which is more than the currently shifted amount. The Monitor Interface shifted left up to the margin only.

User Action: None.

MONS_SHIFTRIGHT, String was found to right of display - use SHIFT RIGHT to see it

Severity: Informational

Explanation: You entered the FIND command and the specified text string was found on the right of the display window. This was because of the text lines being longer than the current width of the display window.

User Action: Enter the SHIFT RIGHT or SET WIDTH command to view the found information.

MONS_SHOWBRKESC, CTRL/*c* is break, CTRL/*c* is escape

Severity: Informational

Explanation: This message indicates the keys you must press to escape back to the calling environment or send a break to the node.

User Action: None.

MONS_SHOWNBKESC, No break is set, CTRL/*c* is escape

Severity: Informational

Explanation: This message indicates the key you must press to escape back to the calling environment and reminds you that there is no break key set.

User Action: None.

MONS_TWOWDW, Already in two window mode

Severity: Informational

Explanation: You entered the TWO_WINDOWS command while the Monitor Interface has already been displaying two log windows on the terminal screen.

User Action: None.

MONS_UPDATING, Updating *node-name*

Severity: Informational

Explanation: You entered the VIEW command and the Monitor Interface is updating the log window to display the console data from system *node-name*. It would take approximately 3 to 7 seconds to complete the update depending on the amount of data to be displayed.

User Action: None.

MONS_USERKEYWAS, *key-name* definition was *string*

Severity: Informational

Explanation: You entered the DEFINE/KEY that overrode a previous key definition that had been done by another DEFINE/KEY command.

User Action: None.

MONS_VCSEVENT, VCS events do not have context

Severity: Informational

Explanation: The current event that is being displayed in the event context window is one of the VCS reserved events which does not have any associated context information.

User Action: None.

MONS_VCSKEYWAS, *key-name* function was *function-name*

Severity: Informational

Explanation: You entered the DEFINE/KEY that overrode a VCS pre-defined key.

User Action: None.

MONS_WAITING, WAIT command in progress, enter CTRL/*c* to interrupt

Severity: Informational

Explanation: The WAIT command is temporarily suspending the Monitor Interface from processing user commands at the **Command:** prompt.

User Action: Enter the Ctrl/**c** key at the keyboard to interrupt the suspension period before issuing any further command.

MSBITORDER, Invalid icon *icon name*, non standard bit ordering unsupported in icon conversion

Severity: Error

Explanation: The specified icon was not converted because the DDIF source image was created with a non standard bit ordering.

User Action: Create icons with graphics editor that saves images with standard bit ordering format.

MULTIPLANE, Invalid icon *name*, multiplane images unsupported in icon conversion

Severity: Error

Explanation: The specified icon was not converted because the DDIF source image was created as a color image.

User Action: Created icon source DDIF files as monochrome images.

NAMESCNT, Too many entries in the names menu

Severity: Warning

Explanation: The number of nodes in the configuration file is more than will fit in the C^3 names menu. The maximum is 32.

User Action: Remove the extra nodes from the configuration file.

NETSUPPORT, Network access for *file-name* is unsupported

Severity: Error

Explanation: VCS locks the configuration during an editing session. Since file locking is not permitted over network access, the file should be on the local system.

User Action: Copy the file to the local system and resume editing session.

NOACTIVE, No active system available

Severity: Informational

Explanation: The VIEW/ACTIVE command was given and there is no node with new data.

User Action: Wait for a node to become active, then issue the VIEW/ACTIVE command. Or, give a VIEW node command for a specific node.

NOAPP, Unable to create application context

Severity: Error

Explanation: The DECwindows application context was not created.

User Action: Try restarting DECwindows and VCS interface, if this does not correct the problem contact software support.

NOBITS, Invalid icon *name*, image contains no pixels

Severity: Error

Explanation: The specified icon contains no bitmap.

User Action: Create non-empty DDIF image file.

NOCCT, Character cell terminal not supported

Severity: Error

Explanation: The C^3 was started without a /DISPLAY qualifier and no default display was defined by the DCL SET DISPLAY command. The C^3 is started as a character cell access interface application in this case. There is no support for a character cell interface for the Central Control Coordinator.

User Action: Use the DCL SET DISPLAY command or the VCS COORDINATE /DISPLAY qualifier to specify a DECwindows server.

NOCOLOR, Invalid icon *icon name*, no color

Severity: Error

Explanation: The Icon Converter was unable to find a region to color in the icon.

User Action: Change icon shape so there is some part of the icon that has an unbroken outline enclosing an inner region.

NOCTXAVAIL, No context records are available for the specified event

Severity: Informational

Explanation: The VCS Monitor Interface determines that there were no context records available for the selected event.

User Action: None.

NODATA, No data available on pipe

Severity: Error

Explanation: Internal software error probably caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

NODECNT, Loaded *n* nodes

Severity: Informational

Explanation: The configuration data for *n* node records was successfully loaded.

User Action: None.

NODEFICON, Cannot fetch the default icon

Severity: Error

Explanation: An error was encountered loading the default node or peripheral icon.

User Action: Quit and restart the C^3 display. Reinstall the original VCS$DW_ICON_DFN.UID file from the VCS kit.

NODELICENSE, System *node-name* is not licensed

Severity: Warning

Explanation: The software license does not have enough units to allow the connection to system *node-name*.

User Action: Contact your system manager and your Digital account representative.

NODEPERIPHDUP, Node *name* has duplicate peripheral *name*

Severity: Error

Explanation: The specified peripheral has already been entered.

User Action: None.

NODEPERIPNOF, Node *name* peripheral *name* not found in configuration file

Severity: Error

Explanation: The peripheral specified must already be entered in the VCS configuration file.

User Action: Add the peripheral to the configuration.

NODEPORTDUP, Node *name* has duplicate port *number*

Severity: Error

Explanation: Each node in a cluster must have a unique port number assigned to it. The specified port is already allocated to another node.

User Action: Re-enter using the correct port number.

NODESCANMISS, Node *name* scan definition missing

Severity: Warning

Explanation: The specified node does not have a scan profile selected. No scanning of events will take place.

User Action: Select a scan profile for the selected node if you want event scanning to occur.

NODESCANNOF, Node *name* scan *name* not found in configuration file

Severity: Error

Explanation: The scan profile for the selected node does not exist in the VCS configuration file.

User Action: Add scan profile to the configuration file.

NODETERMALLOC, Node *name* has terminal line *id* already allocated by *node-name*

Severity: Error

Explanation: Another node already has the specified terminal line associated to it.

User Action: Use another terminal line.

NODETERMINV, Node *name* has invalid terminal line *id*

Severity: Warning

Explanation: The terminal device specified does not exist on the host system.

User Action: Specify valid terminal line on VCS host system.

NODETERMMISS, Node *name* terminal device missing

Severity: Error

Explanation: Each node is required to have a valid terminal line to associated with it.

User Action: Enter the node definition with a valid terminal line.

NODNAMLEN, Node *name* exceeds 6 character limitation

Severity: Error

Explanation: The VCS limitation for the node name length is 6 characters.

User Action: Re-enter with fewer characters.

NODWENTRY, Entry *entry point name* is required but not found

Severity: Warning

Explanation: The access interface application was started with the /INTERFACE=DECWINDOWS qualifier but the required entry point *entry point name* was not found in the application image file.

User Action: If the application does not have a DECwindows interface, omit the /INTERFACE=DECwindows qualifier and start the application again. If the application does have a DECwindows interface, modify and relink the image so that the *entry point name* is included.

NOEFN, Can't allocate required Event Flag resource

Severity: Error

Explanation: The event flag that is used to synchronize the VCS and DECwindows event reporting could not be allocated.

User Action: Quit and restart the C^3 display. Submit a software performance report (SPR) with a copy of your VCS$CONFIGURATION.DAT file if restarting C^3 or VCS does not correct the problem.

NOENTRY, Entry *identifier* is required but not found

Severity: Error

Explanation: The user is attempting to load an Access Interface user application. *Identifier* is one of the required callback routines cannot be found in the shareable image.

User Action: If the entry point *identifier* is missing the accompanying message will be '%LIB-F-KEYNOTFOU'. Verify the callback routine *identifier* is defined correctly in the source and that the image has been linked with an option file that specified this routine name as a universal symbol. If the application image is not installed the accompanying message will be '%SYSTEM-F-PRIVINSTALL'. Use the install utility to install the application. Any other accompanying errors are related failure in the activation of the application image.

NOEVENTS, Cannot View Events

Severity: Informational

Explanation: The VCS Monitor Interface was invoked with the qualifier /NOEVENT. Events (VCS or scan) are not available to this interface.

User Action: Restart the Monitor Interface without specifying the /NOEVENT qualifier to make VCS and scan events available.

NOFILE, Unable to find file *file-name*

Severity: Warning

Explanation: The Monitor Interface cannot find the log file for the time period you specified in the REVIEW command.

User Action: Use the SHOW/DIRECTORY command to determine the periods for which there is log information available. If the period is not available, then the log file for which you are looking was probably archived onto a storage device. Check with the system manager.

NOFONT, Can't load default font

Severity: Warning

Explanation: The default font "Fixed" could not be used to display labels or event text.

User Action: Quit the C^3 display and any other DECwindows applications on the DECwindows server. Restart the C^3 display

first. Restart the DECwindows server software. Try the C³ display on another DECwindows server.

NOFRAME, Invalid icon *icon name*, DDIF image frame not found

Severity: Error

Explanation: No DDIF image frame has been found in the DDIF file specified.

User Action: Make sure the file is a valid DDIF image file; try recreating image file.

NOFUTLOGS, No future data available; specify current or past time

Severity: Error

Explanation: One or more of the dates you specified to the REVIEW command is a future date.

User Action: Change the date to the past or present date.

NOLABELS, C3 window labels can not be displayed

Severity: Warning

Explanation: The initialization of the C³ window labels failed. The C³ display will continue without the labels.

User Action: Take corrective action based on the accompanying message.

NOLICENSE, DEC VAXCLUSTER-CONSOLE use is not authorized on this node

Severity: Fatal

Explanation: The product license has not been registered and loaded on this system.

User Action: Contact your system manager and your Digital account representative.

NOLOGDEV, Logical name VCS$LOG is improperly defined, or missing

Severity: Warning

Explanation: VCS has no access to the directory containing the log file because of an ill-formed or undefined logical name.

User Action: Re-examine the startup procedures to discover why VCS$LOG was not defined to point to the log directory.

NOLOGS, No log files for review found

Severity: Warning

Explanation: No log files were found for the given dates.

User Action: Check the archives for files saved on some other media.

NOMSGFLD, No message fields

Severity: Error

Explanation: Internal software error possibly caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

NONAME, Object has no name

Severity: Error

Explanation: Entity name field is missing.

User Action: Enter name field.

NONESEL, Select one or more icons before choosing a command

Severity: Warning

Explanation: A selection was made from the C^3 Commands menu when none of the icons in the C^3 window were selected.

User Action: Click on an icon to select a node or peripheral. Select a node name from the Names menu to select a node by name. Hold down the shift key to select more than one icon.

NONEXSYS, Non existent system: *node-name*

Severity: Error

Explanation: You specified a system that is not listed in the configuration file.

User Action: Check the VCS configuration information for the list of acceptable system names. Re-enter the command with a system name that has a definition in the current configuration file.

NONEXUS, Non existent user

Severity: Error

Explanation: Internal software error possibly caused by version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

NOPRIV, No VCS privileges

Severity: Error

Explanation: You are not allowed to execute the requested VCS management function.

User Action: If you are in the Monitor Interface, enter the command SET PRIVILEGES and then re-enter the original command again. If you receive this error message, see the system manager to obtain access rights.

NOPRIVILEGES, VCS privileges disabled

Severity: Informational

Explanation: You have entered the command SET NOPRIVILEGES in the Monitor Interface.

User Action: None. You will have to enable your privileges again with the SET PRIVILEGES before you can execute any of the VCS management functions.

NOREQFONT, Unable to load requested font

Severity: Warning

Explanation: The requested font could not be loaded by the server, the default font of the server will be used.

User Action: The server may be low on resources. Try reducing number of applications running on the server at the same time. Also check to make sure the font exists on the server.

NORMAL, Normal successful completion

Severity: Success

Explanation: Function completed successfully.

User Action: None.

NOSCAN, Failed to process scan profile *profile-name* for system *node-name*

Severity: Warning

Explanation: Cannot find profile based on scan name.

User Action: Give the correct scan profile name in the configuration file.

NOSPACE, No spaces allowed in name

Severity: Error

Explanation: The configuration file entity cannot have spaces embedded within the name.

User Action: Remove all spaces.

NOSTART, VCS will not start using this configuration file

Severity: Warning

Explanation: Indicates that VCS will not start using this unverified configuration file.

User Action: Edit the configuration file, issue a verification command to list the problems then fix discrepencies. If the file verifies, the verification flag in the configuration file will be set upon exiting.

NOSTARTLOG, Starting at *date*; No logs prior to this date found

Severity: Informational

Explanation: The Monitor Interface cannot find the log file for the time period you specified in the REVIEW command.

User Action: Use the REVIEW/DIRECTORY command to determine the periods for which there is log information available. If the period is not available, then the log file for which you are looking was probably archived onto a storage device. Check with the system manager.

NOSUCHSYS, Undefined system name

Severity: Error

Explanation: The given system name has no definition in the current configuration file.

User Action: Re-enter the command with a system name that is defined in the current configuration file.

NOTALLCTX, Not all of the context records are available for this event

Severity: Informational

Explanation: The VCS Monitor Interface determines that not all of the context records are available for the selected event.

User Action: None.

NOTEXT, Event text field missing for event *name*

Severity: Error

Explanation: VCS must have a text string associated with an event definition.

User Action: Provide the required text string when adding or modifying the event definition.

NOTFOUND, Not found

Severity: Error

Explanation: The processing of the new configuration file requested by RECONFIGURE has failed. An internal icon data record that is the mate for the current node or peripheral record could not be found.

User Action: Quit and restart the C^3 display. Submit a software performance report (SPR) with a copy of your VCS$CONFIGURATION.DAT file if restarting C^3 or VCS does not correct the problem.

NOWGT, No record for icon

Severity: Warning

Explanation: No internal data record could be found to match the icon that was clicked on.

User Action: Quit and restart the C^3 display. Submit a software performance report (SPR) with a copy of your VCS$CONFIGURATION.DAT file if restarting C^3 or VCS does not correct the problem.

OLDSEC, Attempt to create an existing global section

Severity: Fatal

Explanation: Global sections from a previously running VCS remain on the system.

User Action: Stop all previously running VCS processes and restart VCS.

OPENIN, Error opening *file-name* as input

Severity: Error

Explanation: VCS could not open the requested file for read.

User Action: Check file protection.

OPENOUT, Error opening *file-name* as output

Severity: Error

Explanation: VCS cannot open the configuration file for output. This is usually caused by access protection or none-existent directory structure.

User Action: Check for correct file specification and protection.

OPENREAD, Configuration file locked by another user, opened read only

Severity: Warning

Explanation: Another user is editing the configuration file. VCS only allows one user to make modifications to the file at one time.

User Action: None.

OUTLOCK, Cannot output to node *node-name*, locked by user *user-name*

Severity: Warning

Explanation: You cannot send data to a locked system.

User Action: The SHOW NODES command indicates the owner of the lock. To unlock the node, the owner can release the lock exiting or by using the SET LOCK OFF command.

OVERMAX, Too many defined systems in configuration, maximum is *n*

Severity: Warning

Explanation: The configuration file contains definitions for more systems than VCS supports. The maximum number of systems is displayed in the message.

User Action: Remove as many system definitions as are required to get down to the maximum number of systems.

PARTIALREC, VCS-partial line

Severity: Informational

Explanation: The log data contains a partial line.

User Action: None.

PARTREAD, Partial pipe read, complete data unavailable

Severity: Error

Explanation: Internal software error possibly caused by version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

PERIPHCNT, Loaded *n* peripherals

Severity: Informational

Explanation: The configuration data for *n* peripheral records was successfully loaded. This message will appear twice. The second report includes peripherals that were not included in any node's peripheral list.

User Action: None.

PIPE_CONNECT, Reader connecting to unspecified location in pipe

Severity: Error

Explanation: Internal software error possibly caused by version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

PIPE_RW, Pipe reader tried to write, or vice versa

Severity: Error

Explanation: Internal software error possibly caused by version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

PKTOVER, Data packet overflow

Severity: Warning

Explanation: Internal software error possibly caused by version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

PORTALLOC, The DECserver port cannot be allocated

Severity: Error

Explanation: The DECserver LAT host initiated connection port which was set up and mapped into the corresponding LT device is currently in use or not authorized for access from the VCS.

User Action: Diagnose and correct the problem, then issue the VCS RECONFIGURE command to enable the failed communications path.

PRCQUOTA, The following process quotas may be too small:

Severity: Warning

Explanation: The user is running the Central Control Coordinator and is attempting to initiate another VCS interface. The application has determined that a particular process quota is too low to successfully initiate the interface.

User Action: Run the authorize utility and increase the specified quota. Log back into the system and restart the C^3 interface so that the new quotas can take effect.

PRIVILEGES, VCS privileges enabled

Severity: Informational

Explanation: You have entered the command SET PRIVILEGES in the Monitor Interface.

User Action: None.

PROCCNT, Loaded *n* procedures

Severity: Informational

Explanation: The configuration data for *n* procedure records was successfully loaded.

User Action: None.

PROTOERR, Protocol error

Severity: Error

Explanation: Internal software error possibly caused by version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

RDLINERR, Line error for line *device-name* detected at *date/time*

Severity: Warning

Explanation: While reading data from the indicated console line, an error was detected. This message is followed by a system error message.

User Action: Recovery from this depends on the system error. Refer to *VAX/VMS System Messages and Recovery Procedures Reference Manual* for recovery from these errors.

RDNOTERM, Recording device is not a terminal

Severity: Error

Explanation: The output device you specified is not a terminal.

User Action: Start the recorder again specifying a terminal for an output device.

RECON_COMP, Reconfiguration of VCS complete

Severity: Informational

Explanation: A user has entered the command VCS RECONFIGURE and VCS completes the reconfiguration successfully.

User Action: None. If any of the nodes has been configured out of the current configuration, your Monitor Interface process would be notified by the Scanner in the event summary window.

RECONFIG_DONE, Reconfiguration complete

Severity: Informational

Explanation: The C^3 interface has completed the updates required to run with the new configuration file requested by the VCS RECONFIGURE command.

User Action: None.

RECONFIG_FAIL, Reconfiguration has failed. Continuing with old configuration

Severity: Error

Explanation: An error was detected in the configuration file processing after a RECONFIGURE. The C^3 interface will continue to run with a older version of the configuration file than the other VCS images.

User Action: Quit and restart the failing C^3 displays.

RECONFIG_IGNOR, Configuration files are identical, reconfiguration ignored

Severity: Informational

Explanation: The configuration file that C^3 is using is the same as the configuration file that was selected to run by the VCS RECONFIGURE command. No C^3 configuration file processing is required for this RECONFIGURE.

User Action: None.

RENBADELOG, Renamed bad event log file as *file-name*

Severity: Informational

Explanation: VCS$ELOG notifies that it has encountered a corrupted VCS event log file and recovers from the error by renaming the corrupted file to *file-name* in the directory VCS$LOG before creating a fresh event log file.

User Action: None.

RENBADLOG, Rename bad VCS log file as *file-name*

Severity: Informational

Explanation: VCS$IODL notifies that it has encountered a corrupted VCS log file and recovers from the error by renaming the corrupted file to *file-name* in the directory VCS$LOG before creating a fresh log file.

User Action: None.

REQFAIL, Command for REVIEW interface cannot be generated

Severity: Warning

Explanation: The user is running the Central Control Coordinator and is attempting to initiate the review of one or more nodes. The application tried to format the parameters for the review and failed.

User Action: Change the value of the 'Hours of Log Data To Review' scale in the C^3 Review Customization box and retry the REVIEW.

REVFILE, Reviewing *file-name*

Severity: Informational

Explanation: The specified file is being reviewed on a DECwindows display by the Icon Converter.

User Action: None.

REVIEWQUIT, Exit review by typing CTRL/Z or selecting 'Quit' in the DECterm menu.

Severity: Informational

Explanation: The C^3 review window has reached the end of the review data to be displayed. The review window remains on the

screen so that the user can scroll through the data and cut and paste text into other windows.

User Action: Set the keyboard focus to the REVIEW DECterm window and type Ctrl/Z or pull down the DECterm 'Commands' menu and select 'Quit'.

REVSTATUS, REVIEW as far as *date/time*

Severity: Informational

Explanation: VCS is informing you of the progress of the REVIEW command.

User Action: None.

RICONFERR, Errors during configuration check

Severity: Fatal

Explanation: The node names specified in the VCS RECORD command are not contained in the current VCS configuration.

User Action: Reissue the command, specifying node names that are in the current configuration.

RILINEDIS, System *node-name* has been disabled at *date/time*

Severity: Warning

Explanation: The indicated system has been disabled.

User Action: If you want data from this system, you must ENABLE it.

RILINENAB, System *node-name* has been enabled at *date/time*

Severity: Informational

Explanation: The indicated system has been enabled.

User Action: None.

RINOSYS, No such system *node-name*, system not being recorded

Severity: Warning

Explanation: The node names you specified in the VCS RECORD command are not contained in the current VCS configuration.

User Action: Reissue the command, specifying node names that are in the current configuration.

RIOUT, Error outputing to recording device at *date/time*

Severity: Warning

Explanation: While trying to output to the recorder device, an error was detected. The system error detected is also displayed.

User Action: Recovery from this is dependent upon the VMS error reported.

RULECNT, Loaded *n* rules

Severity: Informational

Explanation: The configuration data for *n* rule records was successfully loaded.

User Action: None.

SAMEBEGEND, Start date:time and End date:time can not be the same

Severity: Error

Explanation: When issuing a REVIEW command, you cannot specify the same start date and time as the end date and time.

User Action: Change the times specified to be different.

SCANCNT, Loaded *n* scan profiles

Severity: Informational

Explanation: The configuration data for *n* scan profile records was successfully loaded.

User Action: None.

SCANCTXERR, Context from console log will not include the event record

Severity: Warning

Explanation: The data scanner is building its internal table of scan event definitions. It has found an event that is defining the context fields in such a manner that the context display will not include the console record that contains the scan event text.

User Action: The context count field should be increased to be at least one greater than the context start field.

SCANDUP, Scan *name* duplicated

Severity: Error

Explanation: There is a scan definition which is duplicated within the scan tree structure.

User Action: Search the scan tree structure, looking for the duplicated event name.

SCANEVEDUP, Scan event *name* duplicated

Severity: Error

Explanation: There is an event definition which is duplicated within the scan tree structure.

User Action: Search the scan tree structure, looking for the duplicated name.

SCANEVENOF, Scan *name* event *name* not found in configuration file

Severity: Error

Explanation: The specified event has not yet been entered in the configuration file.

User Action: Add the event definition before defining the scan profile.

SCANISRECUR, Scan *name* is recursive to *scan-name*

Severity: Error

Explanation: Recursive scan profile definitions are not allowed.

User Action: Find and remove the scan definition which is causing the recursion.

SCANNOF, Scan *name* sub-scan *name* not found in configuration file

Severity: Error

Explanation: The sub-scan must already be entered in the configuration file before adding this definition.

User Action: Enter the sub-scan definition to the configuration before adding this scan definition.

SCANQUALMISS, Scan *name* required event or sub-scan missing

Severity: Error

Explanation: A scan definition must include at least one event definition or another scan definition.

User Action: Re-enter the scan definition using an event or scan definition which already exists.

SELONE, Too many nodes selected

Severity: Error

Explanation: More than one icon was selected when the C^3 'Commands' menu entry 'Connect', 'Show Information' or 'Set Host' was chosen. These commands can only be used with one node at a time.

User Action: Select a single node and choose the command again.

SERVERDUP, Display has duplicate servers *name*

Severity: Error

Explanation: Servers must be unique.

User Action: Remove duplicate server from the list.

SHIFTKEY, Shift key set as *key-name*

Severity: Informational

Explanation: You have entered the command SET SHIFT_KEY in the Monitor Interface.

User Action: None.

SHIFTKEYCLR, Shift key cleared

Severity: Informational

Explanation: You have entered the command SET NOSHIFT_KEY in the Monitor Interface.

User Action: None.

SHUTDOWN, VCS Shutdown

Severity: Informational

Explanation: The VCS$SCANNER, VCS$IODL, and any Recorder and C³ interfaces have been shut down because someone issued a SHUTDOWN VCS command.

User Action: Restart VCS if needed, otherwise, no action is necessary.

SPAWNERR, Error in spawn processing

Severity: Error

Explanation: The C³ Interface is trying to create or communicate with a spawned subprocess. Either the parent or the child cannot perform the desired function.

User Action: Run the Central Control Coordinator application directly from DCL level instead of as a detached process to verify the account has sufficient privileges and quotas to execute. Execute a SHOW SYSTEM command to see what processes exist and what state they are in. Execute a STOP/ID command for any VCS Task. Delete any VCS$SPAWN.COM files that are present. These steps are all part of a normal shutdown of the application, but an abnormal termination may have interfered and left the system in state that will not allow the application to run properly.

SPEEDCHG, Terminal line speed has been changed

Severity: Warning

Explanation: As a result of the reconfiguration process a node's line speed has been changed.

User Action: None.

STRINGTOOLONG, Text string parameter is too long

Severity: Warning

Explanation: The user entered the Monitor Interface OUTPUT command and supplied a text string which is longer than 120 characters.

User Action: Enter the OUTPUT command with a shorter text string. If a text string longer than 120 characters is to be entered, break it up into shorter substrings and use the OUTPUT/NOTERMINATE command to send them to the serviced node.

SUBPRCLM, New subprocess can't be created, a process quota will be exceeded

Severity: Warning

Explanation: One or more of the process quotas is too small to allow a new C^3 subprocess to be created. The quotas that are checked are PRCLM, FILLM, ENQLM and TQLM.

User Action: Take corrective action based on the accompanying message.

SUCCESSVER, Successful verification

Severity: Informational

Explanation: The VCS Configuration file has verified successfully.

User Action: None.

SWERROR, Internal software error

Severity: Error

Explanation: This is an internal software error.

User Action: If no other errors are evident, send in a Software Performance Report (SPR) then ignore the message. If other errors are apparent, restart the failed component.

SYNCHDUPL, Duplicated process *process-name*

Severity: Error

Explanation: You have attempted to start another process *process-name* while one is still active.

User Action: Use the SHUTDOWN command to terminate the current VCS environment and use the VCS$STARTUP.COM procedure to startup another one.

SYNCHINITDONE, *process-name* has completed initialization

Severity: Informational

Explanation: The process *process-name* completes initialization.

User Action: None.

SYNCHINITWAIT, Waiting for *process-name* to complete initialization

Severity: Informational

Explanation: You have entered a user interface before the VCS overhead process *process-name* completes initialization.

User Action: If *process-name* is VCS$IODL and the message SYNCHINITDONE does not follow momentarily, examine the contents of the log file SYS$MANAGER:VCS$IODL_OUTPUT.LOG to determine the cause of the error that halted VCS$IODL.

SYNCHNONOP, *component* is not operational at *date/time*

Severity: Warning

Explanation: The component is either the VCS$IODL or VCS$SCANNER. Either one of the processes has crashed.

User Action: Examine the contents of the appropriate process error log file in SYS$MANAGER to determine the cause. Restart VCS.

SYNCHRESTART, Restart your process to work with the new *process-name*

Severity: Warning

Explanation: The Monitor Interface notifies that the VCS overhead process *process-name* has been restarted and you should restart your interface to resume normal VCS activities.

User Action: Exit the current Monitor Interface and restart it.

SYNCHUNKNOWN, Unknown process synchronization

Severity: Error

Explanation: Internal software error possibly caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification of these images with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

SYNCHWAIT, *component* is waiting for user processes to re-initialize

Severity: Warning

Explanation: *Component* is either the VCS$IODL or the VCS$SCANNER. The VCS$IODL or the VCS$SCANNER is waiting for a previous VCS process to stop.

User Action: None.

SYSGONE, System has been removed from the configuration

Severity: Informational

Explanation: A system that was previously being recorded has been removed from the configuration file and a VCS RECONFIGURE command was then issued.

User Action: None.

SYSLOCKED, System *node-name* has been locked by user *user-name*

Severity: Informational

Explanation: While in the Connect Interface, another user process locked your connected node.

User Action: By leaving the Connect Interface and entering the Monitor Interface, the user can determine the owner of the lock with the SHOW NODES command.

SYSRECERR, Data overrun in recorder interface at *date/time*, data may be lost

Severity: Warning

Explanation: The Recorder Interface, running at a slower speed than the systems being recorded, is losing data.

User Action: The only possible recovery is to use a faster printer for a recording device.

SYSRECONFIG, VCS System has been reconfigured

Severity: Informational

Explanation: The system manager has reconfigured the VCS with the RECONFIGURE command.

User Action: You may have to restart your user interface if the new VCS configuration contains new serviced nodes.

SYSUNLOCKED, System *node-name* is no longer locked

Severity: Informational

Explanation: While in the Connect Interface, another user unlocked your connected node.

User Action: None.

TBLFULL, System data table is full

Severity: Error

Explanation: The internal table of node configuration data is full.

User Action: Examine the configuration file using the Configuration Editor and check the number of node records.

TERMREQ, Termination request received at *date/time*

Severity: Informational

Explanation: VCS$IODL has sent a terminate request to all processes running a VCS interface. This message indicates the reception and compliance with that request.

User Action: None.

TEXTNOTALLOWED, No event text allowed for event priorities 0, 1, 99, 100 - event *event-name*

Severity: Error

Explanation: Events 0,1,99,100 are reserved for VCS events and cannot contain text.

User Action: Change event priority to range 2-98 or remove event text which in effect would be renaming the VCS reserved event.

TIMEOUT, Timed out waiting for completion confirmation

Severity: Error

Explanation: VCS$IODL took longer than 15 minutes to complete and acknowledge a VCS RECONFIGURE or VCS SHUTDOWN command request.

User Action: It is possible that VCS$IODL is behaving abnormally. Start a Monitor Interface with the VCS MONITOR command to ensure that VCS$IODL is working properly. If VCS$IODL is not responding to the Monitor Interface, restart the VCS system using the VCS$STARTUP.COM command procedure.

TOOMANYLEV, Too many levels to specified logical name

Severity: Informational

Explanation: There were more than ten translations required to resolve the logical name.

User Action: Simplify the logical name definition to have fewer than ten translations.

TOOMANYSYS, Too many systems specified in command

Severity: Informational

Explanation: While entering a command, too many systems were specified.

User Action: Re-enter command with less systems specified.

TOOMNYC3, Too many C3 interfaces - Only one per job allowed

Severity: Error

Explanation: A logical name of the form VCS$C3_MBX was found in the job wide logical name table.

User Action: Run the additional C^3 displays in another job.

TRUNCATE, A long line has been truncated

Severity: Informational

Explanation: A console data line was too long to be reviewed.

User Action: None.

TWOWRTR, Pipe has two writers

Severity: Error

Explanation: Internal software error possibly caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification of these images with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

UNSUPLINSP, Unsupported line speed

Severity: Warning

Explanation: Supported line speeds are: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, and 38400.

User Action: Modify the entry using one of the supported line speeds.

USER_NOTINUAF, User *name* not found in User Authorization File

Severity: Warning

Explanation: The user name was entered incorrectly or is missing from the user authorization file.

User Action: Run the VMS utility *Authorize* and verify the presence of the user.

USERCNT, Loaded *n* users

Severity: Informational

Explanation: The configuration data for n user records was successfully loaded.

User Action: None.

USEROVF, Too many users active

Severity: Error

Explanation: One more user processes than VCS$IODL could handle tried to run VCS. This process is not a participant of VCS.

User Action: None.

USERSYSMISS, User does not have any access to *name*

Severity: Error

Explanation: The user is not listed as an authorized user for the specified node or cluster.

User Action: Edit the configuration and add to user definition access to the specified node or cluster.

VERERRNOTSAV, Verification Errors, changes not saved

Severity: Error

Explanation: Verification of configuration file has failed, changes saved to journal file.

User Action: Invoke VCS Editor on the journal file to continue editing session.

VERSKW, Version skew

Severity: Error

Explanation: Incompatible versions of the monitor interface and the .TPU$SECTION files are attempting to run together.

User Action: An invalid section file may have been specified. If the correct section file was specified, a more serious problem may exist. In this case, check the image file identification with ANALYZE/IMAGE. There may be a version skew between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE. These fields should be all the same. If not, reinstall the product or contact software support.

VIRTMEM, Virtual memory running out; REVIEW stopped prematurely

Severity: Warning

Explanation: The amount of data retrieved by a REVIEW command for a user-specified time period, is too large to be safely held in the Monitor Interface. As a result, the review terminated prematurely with only some of the data.

User Action: Save the retrieved data in a disk file, using the WRITE command. Continue the retrieval process by re-entering the REVIEW command using the remaining time period that it could not process in the previous request.

WAITEFN, Error in DECterm controller exit wait

Severity: Warning

Explanation: C^3 failed to set up and start the timer that is used to delay the next request for a DECterm window. The delay gives the DECterm controller time to exit before the next request is issued.

User Action: Take corrective action based on the accompanying message.

WORKING, Working...

Severity: Informational

Explanation: This message is displayed any time a long review is being performed.

User Action: None.

WRTSIZE, Attempt to write pipe unit greater than half pipe size

Severity: Error

Explanation: Internal software error possibly caused by some type of version skews between VCS$IODL.EXE, VCS$IFEX.EXE, and VCS$MSG.EXE.

User Action: Check the image file identification of these images with ANALYZE/IMAGE. These fields should be all the same. If not, reinstall the product or contact software support.

Index

A

Access Interface, 1–6
  Default Application, 5–7
  running the, 5–4
Action routines
  See ENS action routines
ADD, 10–19
  ACTION command, 10–20
  CLUSTER command, 4–4
  EVENT command, 4–6, 10–24
  NODE command, 4–12
  PERIPHERAL command, 4–17
  SCAN command, 4–19
  USER command, 4–22

B

Brdcst action routine, 10–5
Break key, 2–16, 5–10
Break signal, 7–11

C

Central Control Coordinator, 1–5, 2–1, 4–23
  commands, 2–4
  starting, 2–1
  window, 2–2
C^3 Interface
  See Central Control Coordinator
Command file, 5–24
  starting the Monitor Interface with, 2–16
Configuration Editor, 1–6
  character-cell interface, 4–1
  DECwindows interface, 3–1
Configuration Editor commands
  @, 4–2
  ADD CLUSTER, 4–4
  ADD EVENT, 4–6
  ADD NODE, 3–4, 4–12
  ADD PERIPHERAL, 4–17
  ADD SCAN, 4–19
  ADD USER, 4–22
  DELETE, 4–25
  EXIT, 4–27
  HELP, 4–28
  MODIFY CLUSTER, 4–31
  MODIFY EVENT, 4–33
  MODIFY NODE, 4–36
  MODIFY PERIPHERAL, 4–39
  MODIFY SCAN, 4–41
  MODIFY USER, 4–43
  QUIT, 4–45
  SET DISPLAY, 4–47
  SET LOGGING, 4–49
  SET SECURITY, 4–52
  SET VERIFY, 4–53
  SHOW, 4–54

Configuration Editor commands (Cont.)
SPAWN, 4–57
Configuration file, 1–6, 3–1
creating the, 5–22
editing the, 5–22
modifying, 3–24
template, 3–2
CONNECT command, 6–17, 7–5
Connecting to a node, 2–8, 5–10, 6–17
Connect Interface, 1–5
running the, 5–10, 6–17
Console log file
format of, A–3
Context
See Event context
Conversion
icon bitmap, 5–15, 8–1
VCS 1.2 configuration file, 5–15
VCS 1.2 scan file, 5–15
Coordinating VCS interfaces, 5–19
COPY, 10–26
ACTION command, 10–27
EVENT command, 10–30
Copying a buffer into a file, 7–10
Customizing the Central Control Coordinator
DECterm windows, 2–22
from the Configuration Editor, 3–25
general features, 2–13
Monitor Interface, 2–16
Review Interface, 2–17
Set Host, 2–19
Customizing the Monitor Interface, 7–11
CYCLE command, 6–22

D

Data file
ENS, 10–2
Data Scanner, 1–5
DCL commands, 5–1
VCS ACCESS, 5–4
VCS ACCESS /DEFAULT, 5–7
VCS CONNECT, 5–10, 7–5
DCL commands (Cont.)
VCS CONVERT, 5–15, 8–1
VCS COORDINATE, 5–19
VCS EDIT CONFIGURATION, 5–22
VCS HELP, 7–4
VCS MONITOR, 5–24, 7–3
VCS RECONFIGURE, 5–27
VCS RECORD, 5–29
VCS REVIEW, 5–32, 7–9
/EVENT, 7–9
/PREVIOUS, 7–9
VCS SHUTDOWN, 5–38
VCS START LOGGER, 5–39
VCS START SCANNER, 5–40
DECtalk action routine, 10–6
DECwindows interfaces
Central Control Coordinator, 1–5, 2–1
Configuration Editor, 1–6, 3–1
DEFINE/KEY command, 6–10, 6–23, 7–10
Defining
clusters, 3–11, 4–4, 4–31
events, 3–13, 4–6, 4–33
keys, 7–10
nodes, 3–4, 3–25, 4–12, 4–36
peripherals, 3–9, 4–17, 4–39
scan profiles, 3–16, 4–19, 4–41
VCS environment, 3–20, 4–47, 4–49, 4–52, 4–53
VCS users, 3–18, 4–22, 4–43
DELETE command, 4–25, 10–32
Deleting
configuration definitions, 3–25
DISABLE command, 6–28
Disconnect from a node, 7–11
Do key, 6–12
Duplicating
node definitions, 3–25

E

ENABLE command, 6–30
ENS action routines, 10–4
Brdcst, 10–5

ENS action routines (Cont.)
creating, 10–12
DECtalk, 10–6
Mail, 10–5
Window, 10–7
Window1, 10–10
ENS Editor, 10–2, 10–18
ENS Editor commands
ADD ACTION, 10–20
ADD EVENT, 10–24
COPY ACTION, 10–27
COPY EVENT, 10–30
DELETE, 10–32
EXIT, 10–33
HELP, 10–34
MODIFY ACTION, 10–37
MODIFY EVENT, 10–39
SHOW, 10–41
Escape key, 2–15, 5–11
Event context, 2–5, 7–7
Event Logger, 1–5
Event Notification System
See ENS
Events
defining, 3–13, 4–6
scan, 2–5, 3–13, 4–8, 9–26
VCS, 9–30
viewing, 2–5
Event summary window, 6–3, 6–5
@ (Execute Command File) command, 4–2, 6–15
EXIT command, 4–27, 6–32, 7–3, 10–33
Exiting
from Central Control Coordinator, 2–22
from icon review, 8–11

F

FIND command, 6–33, 7–8

G

Getting information
about an event, 2–6
Getting information (Cont.)
about icons, 8–6
about serviced nodes, 7–9
about the configuration, 2–4, 2–10
about the Monitor Interface, 7–9
GOTO command, 6–36

H

Hardcopy terminal
specifying, 5–29
HELP command, 4–28, 6–38, 7–4, 10–34

I

I/O Data Logger, 1–5
Icon colors, 2–2
Icon conversion, 8–1
Interfaces
Access Interface, 1–6
Central Control Coordinator, 1–5
Configuration Editor, 1–6
Connect Interface, 1–5
Monitor Interface, 1–5
Record Interface, 1–6
Review Interface, 1–6

K

Key definitions, 6–10, 7–10

L

Labels, 3–23
Locking a node, 2–8, 2–15, 5–12, 7–11
Log data record, A–14
Log files, 1–6, 2–6
console, 1–7
directory of, 7–9
ENS, 10–4
event, 1–7
reviewing, 7–9
viewing, 5–32
Logical names, 5–1
ENS, 10–4

Log information
current, 7–6
Log window, 6–3, 6–4, 6–5
banner identifier, 6–4
dimensions, 6–4

M

Mail action routine, 10–5
Message buffer, 7–10
Message line, 6–3, 6–5
MODIFY, 10–36
ACTION command, 10–37
CLUSTER command, 4–31
EVENT command, 4–33, 10–39
NODE command, 4–36
PERIPHERAL command, 4–39
SCAN command, 4–41
USER command, 4–43
Modifying
configuration file, 3–24
labels, 3–24
Monitoring a console, 5–24
Monitor Interface, 1–5
command line, 6–3, 6–5
command summary, 7–1
customizing, 7–11
Monitor Interface commands
@, 6–15
CONNECT, 6–17, 7–5
CYCLE, 6–22
DEFINE/KEY, 6–23, 7–10
DISABLE, 6–28
ENABLE, 6–30
EXIT, 6–32, 7–3
FIND, 6–33, 7–8
GOTO FIRST, 6–36, 7–8
GOTO LAST, 6–36, 7–8
HELP, 6–38, 7–4
NEXT EVENT, 6–41, 7–6, 7–7
NEXT SCREEN, 6–43, 7–7
ONE_WINDOW, 6–44, 7–7
OTHER_WINDOW, 6–45, 7–8
OUTPUT, 6–46, 7–5
Monitor Interface commands (Cont.)
PREVIOUS EVENT, 6–49, 7–6, 7–7
PREVIOUS SCREEN, 6–51, 7–7
RECONFIGURE, 6–52
REFRESH, 6–54, 7–7
REVIEW, 6–55, 7–9
/EVENT, 7–9
/PREVIOUS, 7–9
SAVE, 6–61, 7–10
SCROLL, 6–63, 7–7
SELECT, 6–65, 7–5
SET BELL, 6–68, 7–11
SET BREAK, 6–69, 7–11
SET ESCAPE, 6–71, 7–11
SET LOCK, 6–73
SET MOUSE, 6–75, 7–11
SET PRIVILEGES, 6–76
SET SHIFT_KEY, 6–77, 7–10
SET TIME_STAMP, 6–79, 7–11
SET WIDTH, 6–80, 7–8
SHIFT, 6–81, 7–7
SHOW DIRECTORY, 6–83, 7–9
SHOW KEYS, 6–85, 7–9
SHOW MONITOR, 6–87, 7–9
SHOW NODES, 6–89, 7–9
SHOW TIME, 6–92, 7–9
SHUTDOWN, 6–93
TWO_WINDOWS, 6–95, 7–7
UNDEFINE/KEY, 6–96
VIEW, 6–98, 7–6, 7–7
/CONTEXT, 7–7
/EVENT, 7–6
WAIT, 6–101
WRITE, 6–103, 7–9, 7–10
Monitor Interface display
command line, 6–9
controlling, 7–7
event summary window, 6–8
message line, 6–9
one window, 6–3
two windows, 6–5

N

NEXT
  EVENT command, 6–41, 7–6
NEXT EVENT command, 7–7
NEXT SCREEN command, 6–43, 7–7
Node names, 5–1
Node output record
  to indicate text length, A–13

O

ONE_WINDOW command, 6–44, 7–7
OTHER_WINDOW command, 6–45, 7–8
OUTPUT command, 6–46, 7–5
Output terminal
  specifying, 5–29

P

Predefined keys, 5–25
  Monitor Interface, 2–17
PREVIOUS
  EVENT command, 6–49, 7–6
PREVIOUS EVENT command, 7–7
PREVIOUS SCREEN command, 6–51, 7–7
Printing time stamps, 5–29
Processes
  Access Interface, 1–6
  Data Scanner, 1–5
  Event Logger, 1–5
  I/O Data Logger, 1–5

Q

QUIT command, 4–45

R

RECONFIGURE command, 6–52
  See also VCS RECONFIGURE command
Recording data on hardcopy terminal, 5–29
Record Interface, 1–6
REFRESH command, 6–54, 7–7
Remote DECwindows displays, 3–21, 5–5, 5–20, 5–22
REVIEW command, 6–55, 7–9
  /EVENT, 7–9
  /PREVIOUS, 7–9
Reviewing
  icons, 8–5
  logs, 2–10, 7–9
Review Interface, 1–6

S

SAVE command, 6–10, 6–27, 6–61, 7–10
Saving C^3 customization, 3–25
Saving key definitions, 7–10
Scan event record
  to indicate occurrence, A–18
Scan events
  See Events
SCROLL command, 6–63, 7–7
Searching a buffer, 7–8
Security, 3–18, 3–22
SELECT command, 6–65, 7–5
Serviced nodes
  communicating with, 7–5
  defining, 3–4
  Digital systems, 1–2
  standalone VAX, 1–1
  status, 2–2
  third-party, 1–2
  VAXcluster systems, 1–1
Service initiation/termination record, A–10
SET
  BELL command, 6–68
  BREAK command, 6–69, 7–11
  DISPLAY command, 4–47
  ESCAPE command, 6–71, 7–11
  LOCK command, 6–73
  LOGGING command, 4–49
  MOUSE command, 6–75
  PRIVILEGES command, 6–76
  SECURITY command, 4–52
  SHIFT_KEY command, 6–10, 6–77

SET (Cont.)
TIME_STAMP command, 6–79, 7–11
VERIFY command, 4–53
WIDTH command, 6–80, 7–8
Setting host, 2–12
SHIFT command, 6–81, 7–7
/SHIFT_KEY qualifier, 7–10
SHOW
DIRECTORY command, 6–83, 7–9
KEYS command, 6–85, 7–9
MONITOR command, 6–87, 7–9
NODES command, 6–89, 7–9
TIME command, 6–92, 7–9
SHOW command, 4–54, 10–41
SHUTDOWN command, 6–93
SPAWN command, 4–57
Starting
ENS, 10–2
the Access Interface, 5–4
the Central Control Coordinator, 2–1, 5–19
the Configuration Editor, 3–3, 5–22
the Connect Interface, 5–10
the Data Scanner, 5–40
the Event logger, 5–39
the I/O Data Logger, 5–39
the Monitor Interface, 2–9, 2–16, 5–24, 7–3
Status change record
to indicate user status, A–11

T

Template
configuration file, 3–2
Terminal width, 7–11
Testing ENS, 10–4
Time stamps, 7–11
printing, 5–29
TWO_WINDOWS command, 6–4, 6–95, 7–7

U

UNDEFINE/KEY command, 6–96
Unlocking a node, 5–12

V

VAXcluster Console System
overview, 1–1
VCS ACCESS /DEFAULT command, 5–7
VCS ACCESS command, 5–4
VCS CONNECT command, 5–10, 7–5
VCS CONVERT command, 5–15
VCS COORDINATE command, 5–19
VCS EDIT command, 5–22
VCS error record
to indicate an error, A–8
VCS event codes, A–17
VCS event record, A–16
VCS events, 9–30
VCS MONITOR command, 5–24, 7–3
VCS processes
See Processes
VCS RECONFIGURE command, 5–27
VCS RECORD command, 5–29
VCS REVIEW command, 5–32, 7–9
/EVENT, 7–9
/OUTPUT, 7–9
VCS SHUTDOWN command, 5–38
VCS START LOGGER command, 5–39
VCS START SCANNER command, 5–40
Verifying configuration definitions, 3–5, 3–6, 3–7, 3–14, 3–17, 3–21, 3–22, 3–23, 3–26
Version record
to indicate format version, A–6
VIEW command, 6–98
/ACTIVE, 7–6
/CONTEXT, 7–7
/EVENT, 7–6
/HELP, 7–4
/MESSAGE, 7–10
/NEXT, 7–6
Viewing
console and event logs, 5–32
current console logs, 7–6

Viewing (Cont.)
current log information, 7–6
event context, 7–7
events, 2–5, 2–7, 7–6
message buffer, 7–10

W

WAIT command, 6–101
Window1 action routine, 10–10
Window action routine, 10–7
Windows
Central Control Coordinator, 2–2
event, 2–5
WRITE command, 6–103, 7–9, 7–10

Reader's Comments

VAXcluster Console System
Software Manual
AA–GV45D–TE

Your comments and suggestions help us improve the quality of our publications.

Please rate the manual in the following categories:	Excellent	Good	Fair	Poor
Accuracy (product works as described)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Table of contents (ability to find topic)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page design (overall appearance)	☐	☐	☐	☐
Print quality	☐	☐	☐	☐

What I like best about this manual: _____

_____

What I like least about this manual: _____

_____

Additional comments or suggestions:

_____

_____

I found the following errors in this manual:

Page Description

_____ _____

_____ _____

_____ _____

For which tasks did you use this manual?

☐ Installation
☐ Maintenance
☐ Marketing
☐ Operation/Use
☐ Programming
☐ System Management
☐ Training
☐ Other (please specify) _____

Name/Title _____ Date _____

Company _____ Dept _____

Mailing Address _____

_____ Phone _____

Do Not Tear - Fold Here and Tape

digital™

AFFIX
STAMP
HERE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
MRO1–3/L12
P.O. BOX 1001
MARLBOROUGH, MA 01752–9840

Do Not Tear - Fold Here and Tape

digital